Effects of Adaptive Discretization on Numerical Computation using Meshless Method with Live-object Handling Applications

A Dissertation
Presented to
The Academic Faculty

By

Qiang Li

In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mechanical Engineering

Georgia Institute of Technology
May 2007

Effects of Adaptive Discretization on Numerical Computation using Meshless Method with Live-object Handling Applications

Approved by:

Dr. Kok-Meng Lee, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Shreyes Melkote
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Bruce Webster
Department of Poultry Science
University of Georgia

Dr. Wenjing Ye
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Chen Zhou
School of Industry and System Engineering
Georgia Institute of Technology

Date Approved: November 13, 2006

To my family.

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to a number of people who have helped or involved in this thesis research. First of all, I would like to thank my advisor Dr. Kok-Meng Lee. His advices were always stimulating and helpful when I was facing difficulties in my research. Furthermore, his enthusiastic and tireless work attitude has taught me what the professional work ethic is. Without his constant encouragements and supports, this research could not have been done. Secondly, I would like to acknowledge my other thesis committee Dr. Shreyes Melkote, Dr. Wenjing Ye, Dr. Bruce Webster and Dr. Chen Zhou for taking time to review my proposal and dissertation, and provide me their valuable suggestions and comments.

The author deeply appreciates the staffs of Georgia Tech Research Institute especially Garry McMurray, Debao Zhou, Sean Thomas, Jonathan Holmes and Wiley Holcombe for their help and support on Intelligent Live Object Transfer project. The appreciation also goes to Dr. Wayne Daley for his suggestions and encouragements.

I also want to thank my former team member Dr. Chao-Chien Lan, Jeffry Joni and Dr. Xuecheng Yin, and fellow students Mati Chessin and Shaohui Foong for their help on Live Object Transfer project. I especially have to thank Ziyen Ng for his help on fabricating the flexible grasper and Hungsun Son for his help on fabricating the electromechanical actuator presented in this thesis. Thank you all for your help, discussion and friendship.

Finally, I want to give my special thanks to my wife Jingying Zhang, my parents Jizhou Li and Yanfang Chen, my brother Hua Li and my daughter. Only with their patient and support, I can complete this thesis.

This research is jointly funded by the US Poultry and Eggs Association, and the Georgia Agriculture Technological Research Program (ATRP).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Upper case	Descriptions
B	Magnetic field flux density
C	Material compliant tensor
F	Force vector
H	Magnetic field intensity
J	Current density
M	Material magnetization
P	1st Piola-Kirchhorff (PK) stress tensor
S	2nd Piola-Kirchhorff (PK) stress tensor
T	Maxwell stress tensor
X	Vector of material coordinate
E	Young's modulus
G	Shear modulus
G_p	Virtual works by contact force
I	Moment of inertia
J	Jacobi coefficient
N	FEM shape function
W_i	Virtual internal works
W_e	Virtual external works
X	Material coordinate

Lower case	Descriptions
b	Body force
n	Normal vector
t	Tangential vector
u	Displacement vector
x	Coordinate vector in deformed state
d	Nodal control value
g_n	Normal gap function
g_t	Tangential gap function
i	Index
j	Index
k	Index
k_n	Normal penalty parameter
k_t	Tangential penalty parameter
t_i	Surface tensor
t	Time
v	Velocity
x	Coordinate in deformed state
y	Coordinate in deformed state
z	Coordinate in deformed state

Greek	Descriptions
Λ	Kernel function
Ψ	Reproducing kernel basis function
$\tilde{\Psi}$	Modified reproducing kernel basis function
Φ	Unknown field variable
Θ	Function with a discontinuous first-order derivative at $\boldsymbol{x}_a$
Γ	Material boundary
Ω	Computational domain
μ_0	The permeability of the free space
μ_r	The relative permeability of the material
μ	Friction coefficient
σ	cochy stress
θ	Angle
ε	Green strain tensor
ω	Angular velocity
ρ	Material density
ν	Poisson's ratio
η	Natural coordinate
ζ	Natural coordinate
α	Angle
β	Angle
γ	Angle
τ_{cn}	Contact force in normal direction
τ_{ct}	Contact force in tangential direction
ξ	Damping ratio

LIST OF ABBREVIATIONS

Abbreviation	Descriptions
1D	One dimensional
2D	Two dimensional
3D	Three dimensional
BEM	Boundary element method
BVP	Boundary value problem
DAE	Differential algebra equations
FDM	Finite difference method
FEM	Finite element method
IE	Integral equations
ML	Meshless
MLM	Meshless method
ODE	Ordinary differential equations
PDE	Partial differential equations
SFF	Strong form formulation
WFF	Weak form formulation

SUMMARY

Numerical methods that are more reliable, general and stable have become increasingly popular in industry. As the most widely applied engineering computational method, the Finite Element Method (FEM) has difficulty solving certain problems where its mesh has to be modified during the computation. In this research, we focus on a new computational method called the Meshless Method (MLM). This method is built upon the same theoretical framework as FEM but needs no mesh. Consequently, the computation becomes more stable and the adaptive computational scheme becomes easier to develop.

The major issue associated with MLM is its lower computational efficiency compared with FEM. Adaptive computations can help reduce the number of nodes used in computation and improve the efficiency. For this reason, this research investigates practical issues related to the MLM and develops an adaptive algorithm to automatically insert additional nodes and improve computational accuracy. The study has been in the context of the two engineering problems: magnetic field computation and large deformation contact. First, we investigate the effect of two discretization methods (strong-form and weak-form) in MLM for solving linear magnetic field problems. Special techniques for handling the discontinuity boundary condition at material interfaces are proposed in both discretization methods to improve the computational accuracy. Next, we develop an adaptive computational scheme in MLM that is comprised of an error estimation algorithm, a nodal insertion scheme and a numerical integration scheme. As a more general approach, this method can automatically locate the large error region around the material interface and insert nodes accordingly to reduce the error. We further

extend the adaptive method to solve nonlinear large deformation contact problems. Contact problems are time-consuming to solve since they are highly nonlinear problems and often need a lot of iterations to converge. With the ability to adaptively insert nodes during the computation, the developed method is capable of using fewer nodes for initial computation and thus, effectively improves the computational efficiency.

Engineering applications of the developed methods have been demonstrated by two practical engineering problems encountered in the development of the live object transfer project at Georgia Tech. In the first problem, the MLM has been utilized to simulate the dynamic response of a non-contact mechanical-magnetic actuator for optimizing the design of the actuator. In the second problem, the contact between the flexible finger and the live poultry product has been analyzed by using MLM. These applications show the developed method can be applied to a broad spectrum of engineering applications where an adaptive mesh is needed.

CHAPTER I

INTRODUCTION

1.1 Motivation and Background

Due to advances in computational technologies in past two decades, many numerical simulation tasks, which were once considered computationally formidable or could be addressed only by a supercomputer, can now be carried out by a desktop computer. Encouraged by this trend, more and more research effort has been devoted to developing numerical tools to facilitate the design or analysis of engineering systems. These efforts, in turn, have led to more reliable, more powerful and faster software packages for numerical simulation. In engineering designs, numerical methods along with high-fidelity mathematical models are able to predict the behavior of an engineering system before the physical system has been built. This drastically reduces the number of different configurations for experimental investigation and thus saves the cost and time in design. In many situations, numerical simulation can effectively reduce or replace expensive experimental studies as a primary investigation tool for engineers.

In this research, a numerical computational tool is developed for solving engineering problems encountered in the development of an automation system for handling of live objects [1-5].

Two particular engineering problems have motivated this thesis research. The first is the need to develop actuators for manipulating the orientation of live objects. For this, electro-mechanical actuators without mechanical contact have been proposed for use in

order to reduce mechanical noises and vibration and therefore to minimize the reaction and unwanted motions from live objects. The second problem is to develop a method to analyze the mechanical contact between grasping fingers and the body of live objects. Contact involving deformable bodies is a highly nonlinear mechanical problem as the contact location, geometry and contact forces are not known before the problem is fully solved. It becomes even more complicated when live objects are handled since highly compliant fingers capable of large deflection must be used to avoid damaging the objects. Previous research has led to a means to locate the contact based on a beam model [6]. Although contact models based on simple beams are efficient, they only provide overall contact force information, as the object being grasped is treated as a rigid body. Furthermore, the beam model becomes inappropriate when the contact point is close to the clamped end of the finger.

1.2 Background

It is desired to have a common numerical tool for solving both electromagnetic fields and displacement fields. Lumped parameter modeling methods are commonly used for design analysis of robotic systems. However, they have difficulty computing the detailed information about distributed fields needed for the preceding two problems. Other mathematical models capable of predicting distributed fields include differential algebraic equations (DAE), integral equations (IE) and partial differential equations (PDE) etc. Among them, partial differential equations has been one of popular models, as it is the natural description of a wide class of physical systems, including solid mechanics, fluid mechanics, heat transfer and electromagnetic field etc. Although PDE models have

existed for centuries, analytical solutions for most PDE systems are only available for a few special situations. The advent of electrical digital computers in the 1940s made solving PDE system numerically possible and since then, many numerical methods for solving PDE have been proposed. Common methods include finite element method (FEM), boundary element method (BEM) and finite difference method (FDM). FEM has been the most widely adopted among these methods, and commercial FEM software packages are widely available today.

The popularity of FEM is mainly due to its generality and robustness. Unlike other methods such as FDM and BEM, the analysis procedures in FEM (which normally consist of pre-process, solve and post-process) are standard routines, and need almost no additional formulation for the computation in general. This feature helps reduce the cost and time spent on learning the method. In addition, as the human involvement in the analysis is reduced, the error due to human mistake can be minimized. After more than half a contrary of development, FEM packages have become so easy to use that an engineer with little knowledge of FEM theory can easily perform some basic FEM analyses.

Even with such a success, there is room for improving FEM. Among the issues, one of the essential tasks in a FEM analysis is the generation of a FEM mesh. The FEM mesh, which can be various shapes such as triangular or rectangular, is used to discretize the physical geometry. Figure 1-1 shows one FEM mesh example [7].

While the FEM mesh enables it to handle complicated geometry, the mesh structure also contributes to a number of problems:

1) It is difficult to generate and modify the FEM mesh due to its interconnected structure.

2) With the stringent shape requirements, the strongly distorted mesh (such as may arise for example in large deformation problem) breaks down the simulation.

3) The computational accuracy of FEM significantly depends on the quality of the mesh.

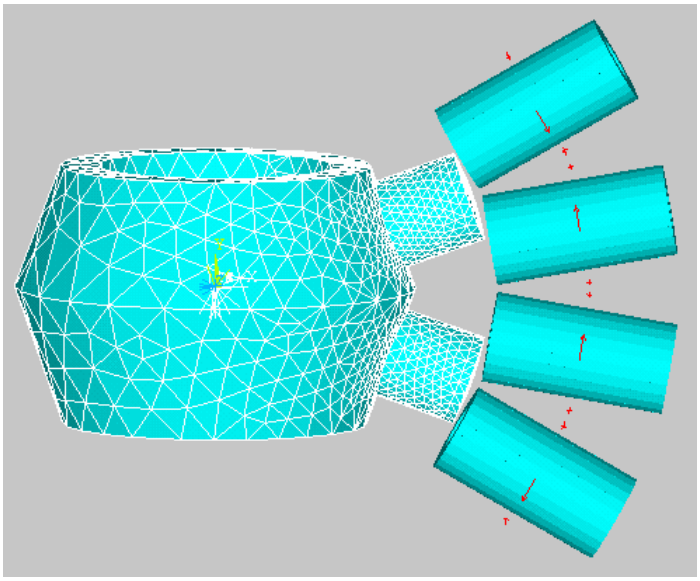


Figure 1-1 A example of Finite Element mesh [7]

Due to the above difficulties of the FEM mesh, our previous effort to solve the preceding two problems encountered in developing an automotive live-bird transfer using FEM has been less than optimal. Excessive deformation of flexible finger caused severe element distortion and subsequently the breakdown of the simulation; re-meshing is often needed in order to resume the computation. This problem becomes especially serious for motion simulation, since the unknowns (contact geometry and forces) change with both location and time. The remesh is often done manually and is prohibitively time-consuming. Similar remesh problems are also found in the computation of electromagnetic fields for designing non-contact actuators. The gradient of

electromagnetic fields around an electromagnetic source or sink is much higher than other areas such as free space; a mesh that helps provide the details of the field in such a region (while keeping the number of global nodes as little as possible) is desired. This requirement is a challenge for designing an effective mesh generation algorithm.

Figure 1-2 shows an example problem of mesh distortion. This example simulates a rubber block under compression using Ansys, a commercial FEM package. Due to the excessive mesh distortion, the simulation eventually fails when the rubber is compressed to less than half of its original length. Even though considerable effort has been devoted to improving the design of the mesh and the algorithm to generate it, generation of the proper element structure remains a significant challenge; human involvement is still unavoidable for most engineering analyses with FEM.

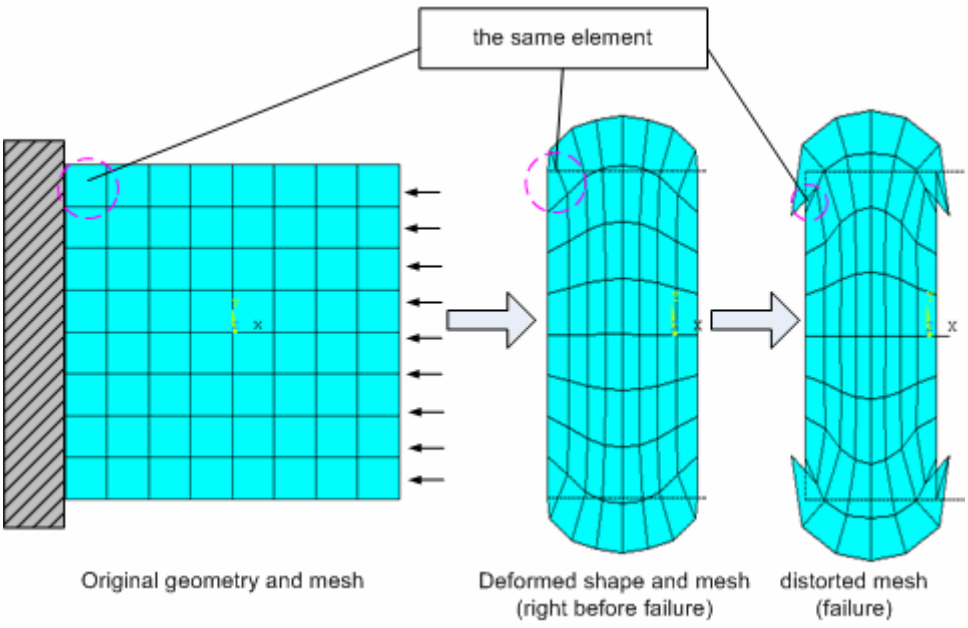


Figure 1-2 A mesh distortion example in FEM

Recently meshless methods (MLM) [8-15] that inherit many advantages of FEM have emerged and yet, they need no explicit mesh structure to discretize geometry. As a result, it greatly reduces the difficulty associated with FEM meshes. Since MLM does not need the element structure to discretize the geometry, the mesh generation algorithm only needs to deal with nodes rather than elements. This important feature makes MLM an attractive alternative for solving engineering problems (such as large deformation contact and fracture mechanics) where the adaptive meshes are often needed.

In this research, we develop a meshless method that is able to solve general boundary value problems (BVP) such as large deformation and mechanical contact and magnetic fields. The method formulated here for design of a flexible grasper uses general governing equations of continuum mechanics to provide more precise stress information. The solution provides a useful means to assess the injury of live birds that are handled in production. This same method can also be used to obtain a detailed magnetic field description that can be used to compute the magnetic force for design of the actuator. While the MLM is developed in the context of live-object handling, we expect that the method will have a broad spectrum of engineering applications.

1.3 Reviews of Prior and Related Works

This section is divided into four parts. The first part reviews general numerical methods for solving BVP. In the second part, the development of the meshless method is revisited. In the last two parts, we discuss some previous research publications on the two engineering problems being considered in this thesis: mechanical contact and magnetic field computation.

1.3.1 Numerical Methods for Solving Boundary Value Problem

Numerical methods are often used to solve BVP's, particularly when closed form solutions are not available. These methods include FDM, FEM, and BEM.

FDM discretizes the governing PDE directly using their strong form. Although it is the most straight forward way to obtain the discrete system equations, it is difficult to handle boundary conditions with FDM. For a problem domain with complex geometry, the discretization of the geometry and the application of the natural and essential boundary conditions can seldom be done automatically by a computer program with no human involvement. As an example, Figure 1-3 shows a FDM grid for simulating the manufacturing process to draw optical fiber [16]. For a complicated geometry like this, additional efforts have to be spent on deriving a customized curvilinear coordinate to transform the geometry to a regular shape.

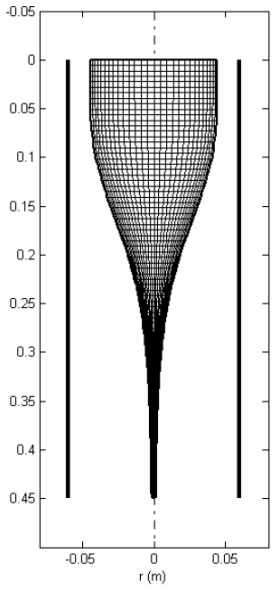


Figure 1-3 An example FDM grid for an irregular shape of geometry [16]

Finite Element Method

FEM, a commonly used method with abundant software available commercially, has a history of over half a century. FEM differs from FDM in many ways:

1) FEM uses the weak form of the governing PDE's.

2) FEM has a variety of types of element; its shape functions are built upon the element.

3) It discretizes the problem domain with a highly automatic mesh generation program. Boundary conditions can be easily applied once the mesh generation is done. However, the pre- and post-processes of the computed setup always play an important role for a good FEM program.

Since FEM discretizes the whole physical domain into small elements, its computation could fail catastrophically due to the singularity of elemental Jacobian matrix when the element experiences large deformation. In order to alleviate this problem, Arbitrary Lagrangian Eulerian (ALE) formulation [17, 18] was proposed. The basic idea of ALE formulation is to make FEM mesh relatively independent of the material so that the mesh distortion can be minimized. Figure 1-4 shows an ALE mesh example [19]. However even with ALE formulation, large deformation can still cause severe numerical errors in many cases.

Another typical problem of FEM mesh is related to the change of physical geometry in the computational process, which often happens in problems such as crack propagation. Many adaptive mesh generation algorithms have been proposed for solving such problems. Figure 1-5 [20] shows one example of adaptive mesh to simulate the trajectory of the crack propagation in a rectangular material with two holes. Nevertheless,

dynamic reconstruction of an FEM mesh in the computational process always poses a significant challenge to a design due to the complexity of the FEM mesh.

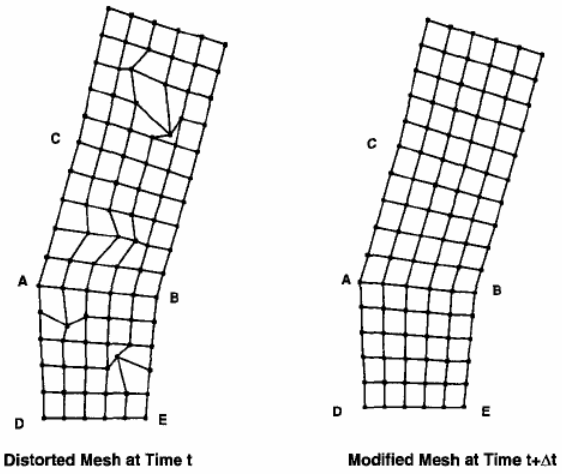


Figure 1-4 An Example of ALE mesh [19]

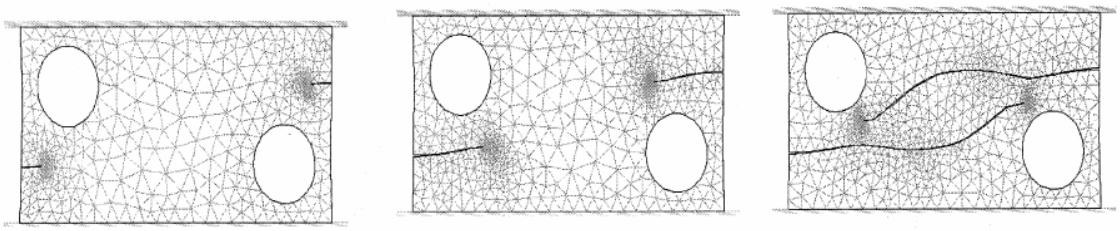


Figure 1-5 Adaptive mesh generation in a crack propagation simulation [20]

Overall, meshes in FEM make it easy to use but a high quality mesh is often essential to ensure accurate solutions; thus, mesh generation for FEM remains an important research topic. As computer speed has grown, in certain extreme cases, the effort devoted to constructing an appropriate mesh has exceeded the effect for solving the problem itself.

BEM, unlike FEM, only needs to discretize the boundary of the problem domain and hence, can reduce the computational time significantly as compared to FEM. The

disadvantages of BEM lie on the difficulty of formulating and solving nonlinear problems, and the need to have knowledge of the suitable fundamental solution.

1.3.2 Development of Meshless Methods

Recently, considerable research has been devoted to the development of meshless methods (MLM) for solving BVP [8-15]. As compared to its counterparts such as FEM, MLM has some advantages:

1) It discretizes the physical domain into a scattered set of points and uses shape functions to interpolate the field variables at a global level.
2) The fact that MLM does not need explicit meshes greatly reduces the dependency on a mesh generation program.
3) In addition, computed results using MLM are generally smooth; therefore, it requires no post-processing as often needed in FEM.

The first MLM, known as the smooth particle hydrodynamics method, was developed in the late 1970's [21-23] but it did not attract much attention until the 1990's. Nayroles et al. [24] developed the diffuse-element-method for structural analysis, and it was later improved leading to a relatively complete element-free-Galerkin-method [12]. After that, many MLMs were proposed, such as the reproducing kernel particle method (RKPM) [15, 25], the natural element method [14, 26], the free mesh method [27], the finite spheres method [28], the local Petrov-Galerkin method [8] and the h-p cloud method [10]. The basic converging theory of MLM was also established by [10, 13]. On the basis of these proposed methods and the theoretical development, MLM has been

applied to many engineering areas, such as fracture mechanics [29-47], fluid mechanics [48-70], MEMS [71-76], and electromagnetic computation [77-85].

The basic formulation of MLM can be generally divided into two categories – strong-form formulation and weak-form formulation. Most of the current MLM applications have been based on the Gelerkin (global weak-form) formulation. Gelerkin based MLM is similar to FEM in that they both require numerical integration to form the disrectized system equations. However, unlike FEM where the basis functions are simple piecewise polynomials, the basis functions used for MLM are often highly nonlinear and not in closed form, as they must satisfy a number of stringent requirements [86].

Some commonly used methods for generating the basis functions include the moving least squares method [12], the reproducing kernel particle method (RKPM) [86], point interpolation method [87] and the natural element method [14]. In general, Gelerkin-based MLM requires higher-order numerical integration and a background mesh (unlike the mesh in FEM, it is independent of the nodes) for the global integration, which tends to increase the computational cost. In addition, most of the basis functions in MLM do not have interpolation property, which often makes direct application of the essential boundary conditions difficult. More recently, the Petrove-Galerkin method [8] has attracted some interest as the formulation uses a local integration scheme (local weak form) requiring no background mesh. However, the drawback of the Petrove-Galerkin method is the difficulty in handling of the numerical integration near the boundary and the asymmetry of the discretized system matrix.

An alternative to eliminate the difficulty of handling boundary conditions at the interface and to avoid numerical integration in deriving the discretized system equations

for MLM is to use another formulation – the strong form of the governing equation. Strong form formulation has been commonly used in FDM and is popular in analyzing fluid mechanics. Several MLM's formulated using the strong form of governing equation have been successfully applied to analyzing mechanics problems; notably, finite point method [88] and the point collocation formulation (PCF). More recently, the point collocation method has been proposed for electromagnetic field analysis [78]. Unlike the Galerkin formulation, PCF uses the strong form of the governing equations to directly obtain a system of discretized equations without numerical integration. For linear problems, PCF appears simpler and requires less computational time. Additionally, it is easy to add nodes to improve computational accuracy at any desired local area.

1.3.3 Review on Magnetic Field Computation

In the past, many methods have been developed for solving magnetic field problems. These methods can be roughly divided into two categories; the integral point of view, and boundary-value point of view. Methods based on the integral point of view [89], such as image method [90, 91], require little computation but can only be applied to linear problems. Methods from the boundary value point of view include FEM [92-96], FDM [97-99], BEM [100-104] and MLM [77-84] which are general numerical methods for solving the BVP that have been reviewed in previous sections. In the computation of magnetic fields, FEM has been the most general method, as it can handle complicated geometry and material nonlinearity.

According to FEM convergence theory, reducing the nodal spacing will improve computational accuracy and convergence. However as mesh density increases, computational efficiency reduces. Hence a high mesh density should not be used in whole

computation domain but in only areas where the gradient of magnetic field is high. Current mesh generation programs can seldom generate satisfactory mesh in a totally automatic manner due to the element structure required by FEM. This is time-consuming, particularly for solving dynamic problems or in situations when a lot of configurations must be examined to determine an optimal design.

Recently, the meshless method has attracted attention in the community of computational electromagnetics due to its potential for being a more stable computational method. Some MLM researchers [77, 80, 81, 83-85] utilized the weak form formulation to compute electromagnetic fields while others [78] employed the strong form formulation. While progress has been made in applying the meshless method to electromagnetic computation, most of these studies have focused on simple test problems; the advantage and disadvantage between different formulations need to be further investigated. Also many practical issues are yet to be solved, which include handling the discontinuity of magnetic fields at material interfaces.

1.3.4 Review on Mechanical Contact Computation

Mechanical contact problems are highly nonlinear even for cases involving small deformation as the contact location, geometry and force are not known before the problem is fully solved. Past effort for this problem has led to two methods of formulation; variational inequality [105-108] and variational equality [109-112]. The method of variational inequality solves the problem fundamentally using a rigorous mathematic point of view; the focus has been on the proof of existence and uniqueness of solution. This mathematical method has been successfully applied to some problems with

frictionless contact [107]. Its applicability to frictional contact has been found to be difficult.

The method of variational equality formulates the problem along with a numerical scheme such as FEM or BEM. Although the variational equality method is not as mathematically rigorous as the variational inequality method, this approach has some engineering successes [110]. With rapid increases in computational power, the approximation of a true solution has been found to be generally acceptable [110]. The three most commonly used variational equality formulations are the Lagrange multiplier method, the penalty method, and the augmented Lagrange method. The Lagrange multiplier method needs to add new variables besides FEM nodal unknowns, which tends to increase the computation load. The penalty method is relatively easy to formulate and has a clear physical interpretation, but it only satisfies the contact constraint approximately. In addition, it is often difficult to choose a proper value for the penalty parameter as too small a value will render the accuracy of the result unacceptable. On the other hand, if the value are too large the system could become ill-conditioned [113]. The augmented Lagrange multiplier method combines the advantage of the previous two methods. However its computation is more complicated.

A common feature of all the above three methods is the way they impose the contact constraint through discretized nodes. It has been shown that increasing the node density around the contact area improves the accuracy significantly [114]. Considerable effort has been devoted to developing good adaptive FEM algorithms for contact problems [114-118]. However, the time need to remesh is still not widely accepted for

FEM in many applications. Figure 1-6 shows the evolution of adaptive FEM mesh in a steel-rubber contact simulation [118].

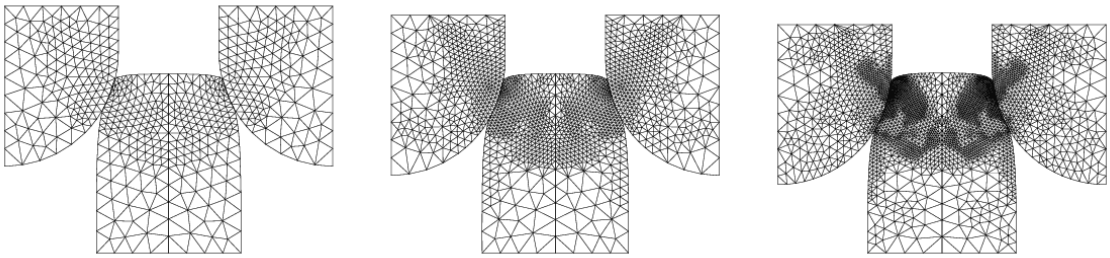


Figure 1-6 FEM mesh in a steel-rubber adaptive contact simulation [118]

1.4 Research Objectives and Tasks

As mentioned previously, MLM has a number of advantages making it a promising numerical method for solving certain difficult engineering problems that need an adaptive mesh. Most previous researches in MLM focus on proposing new methods for constructing basis functions. Little attention has been put on solving practical applications which exploit the advantages of MLM. Motivated by the live-object handling application, this thesis research focuses on analyzing the effect of adaptive discretization on the performance of MLM.

1.4.1 Objectives

This thesis research has three specific objectives.

- The first objective is to study the effects of different discretization methods on MLM.

- The second objective is to develop an adaptive discretization scheme to improve computational accuracy of MLM at any desired local area without sacrificing the accuracy in the overall solution.
- The third objective is to illustrate its application to practical engineering problems encountered in the development of a live-object transfer system.

1.4.2 Tasks

Although it is relatively easier to add nodes in MLM, difficulties (such as the appropriate assignment of the parameters to these additional nodes and the estimation of the corresponding errors) must be overcome before the adaptability of MLM can be practically applied. In order to solve this problem, this thesis research is organized into three tasks:

Task 1: This task is to study different ways of discretizing the continuous mathematical model for meshless methods along with an adaptive discretization method.

Task 2: This task aims at validating the discretization methods developed in Task 1, and investigating their effects on the computation of magnetic fields.

Task 3: In this task, the above methods in Task 1 and 2 are employed to solve a mechanical contact problem.

1.4 Summary of results

This research investigates practical issues related to the MLM and develops an adaptive algorithm in the context of solving two engineering problems: magnetic field

computation and large deformation contact. Specifically, the contributions of this research are briefly summarized as follows:

1) We develop a relatively complete adaptive meshless computational scheme for solving linear BVP based on the weak form formulation. In this computational scheme, we provide the solutions for three major problems in developing the adaptive method; namely, error estimation algorithm, nodal insertion scheme and numerical integration cell reconstruction. The application to the linear magnetic field computation has demonstrated that the developed method is easy to apply in practice, and is able to faithfully locate the large error region and to insert additional nodes accordingly to improve the computational accuracy.

2) We extend the meshless method to solve for the magnetic field and the solution to contact problems involving large deformation. The developed methods are validated by comparing the obtained solution to the analytical solution, and can serve as a basis for future development of more complete design tools for solving engineering problems where an adaptive mesh is needed.

3) We develop methods to handle the boundary condition that involves an interface of two different magnetic materials. Both meshless discretization methods, strong form formulation and weak form formulation, are considered. We investigate the effect of proposed methods for linear magnetic field computation. The computational results show the developed methods are able to handle the magnetic field discontinuity and successfully improve the computational accuracy.

4) We develop a general formulation for solving a mechanical large deformation contact problem in MLM based on sliding-line algorithm and penalty method. We extend the

previous linear adaptive meshless method to solve nonlinear problems. Specifically, the effect of rigid body motion can be eliminated and large error regions can be identified.

5) We demonstrate an engineering application of the developed methods with two practical problems encountered in the development of an automation system for transferring live objects. The first application is to design a non-contact mechanical-magnetic actuator to manipulate the orientation of birds. The second example is to analyze grasping contact for flexible finger grasper.

1.5 Organization of This Dissertation

The remainder of the dissertation is organized as follows.

In Chapter II, we begin with a brief review of the methods of constructing a meshless basis function. In this regard, the reproducing kernel method is used as an example. The formulations of two discretization methods (strong-form and weak-form) are subsequently presented. The comparisons of the two methods along with FEM are made in terms of their computational speed and accuracy by using a Poisson-like problem. Based on the results, the advantages and disadvantages of these two methods are analyzed.

In Chapter III, methods for handling the field discontinuity in both strong-form and weak-form formulation are presented. Numerical examples are given to illustrate the effects of the proposed method for magnetic field computation.

Chapter IV presents the linear adaptive ML computational scheme with application to magnetic field computation. Three important procedures, error estimation,

nodal insertion and numerical integration, are explained in detail using one dimensional or two dimensional numerical examples.

The adaptive MLM in Chapter IV is extended to solve a nonlinear mechanical contact problem with large deformation in Chapter V. This chapter begins with the meshless formulation for mechanical large deformation contact. They followed by the modified formulation of the error estimation. Validation and engineering applications are presented in the last part of this chapter.

In Chapter VI, the developed MLM is applied to solve the two practical engineering applications encountered in the development of the automation system for transferring live object; non-contact mechanical-magnetic actuator, and mechanical contact for flexible finger grasper.

Finally, in Chapter VII, the conclusions of the research are summarized and recommendation and possible future development of this research are provided.

CHAPTER II

BASIS FUNCTIONS AND DISCRETIZATION METHODS

2.1 Overview

As in many other numerical methods, meshless methods convert the continuous model of physical system (usually PDEs) into discretized form in order to solve the problem numerically. We refer to such procedures as discretization methods. These discretization methods often have significant effects on computational efficiency and accuracy. The discretization methods used by FEM and MLM are very similar in the sense that they both approximate the continuous model with individual nodes and a basis function. As mentioned previously, FEM and MLM are often said to be built upon the same theoretical frame work partially due to the fact that they share similar discretization methods. However unlike FEM where the construction of the basis function relies on the mesh structure, the basis functions in most meshless methods are independent of meshes. Because of this, the MLM is able to overcome the limitations due to the mesh. As one of the essential components in the meshless method, ML basis functions have been studied extensively in the past decade and many basis functions have been proposed. Figure 1-1 compares the solution procedure of FEM and MLM.

The remainder of this chapter is organized as follows:

1) The method for constructing a meshless basis function is illustrated with the reproducing kernel method chosen as an example. Some of the important properties of the ML basis function are reviewed.

2) The formulations of two different discretizations, (weak-form and strong-form) are presented and illustrated with a Poisson-like problem as an example. The differences between the two methods are analyzed and summarized.

3) The convergence speed and computational time of the two meshless formulations are investigated and compared against the solution computed using the finite element method.

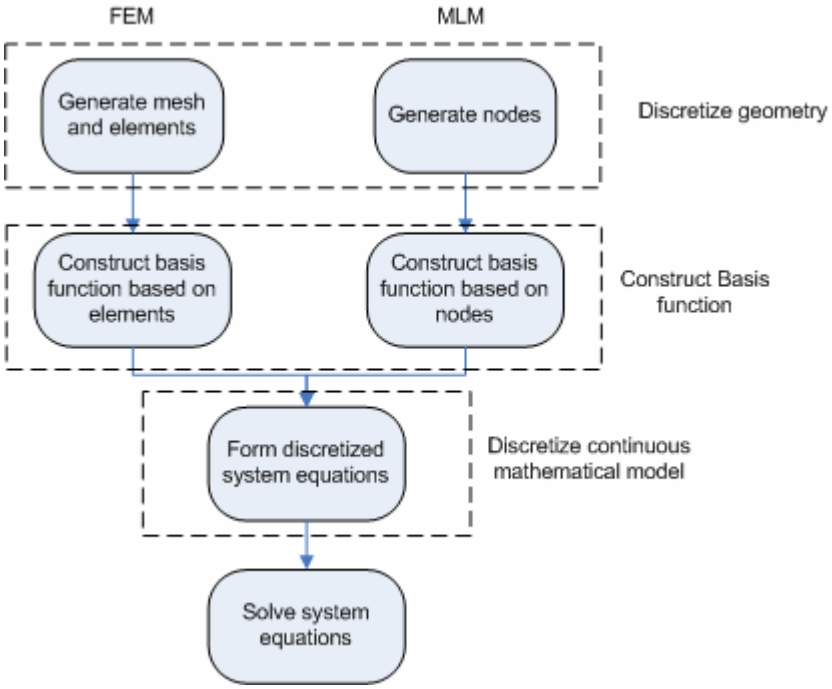


Figure 2-1 Solution procedure of FEM and MLM

2.2 Meshless Basis Function

As shown in Figure 2-1, in the solution procedure of the MLM, the ML basis function must be constructed after the nodes are generated. Without relying on elements, the construction of the basis function in MLM is solely based on the relationship among nodes. The methods for constructing basis functions are very important in the sense that

they have direct effects on the efficiency and accuracy of the solution. In this section, the reproducing kernel method [119] is used as an example to illustrate the procedure for constructing the ML basis function. This method was developed from the earlier smooth particle hydrodynamics method (SPH) which, in turn, was based on the idea of kernel approximation. Thus, we begin here with the kernel approximation in the following section.

2.2.1 Kernel Approximation

The kernel estimation of the function Φ(x) can be written as

$$\Phi_a(\mathbf{x}) = \int_{-\infty}^{\infty} \Lambda(\mathbf{x}-\mathbf{x}')\Phi(\mathbf{x}')d\mathbf{x}' \tag{2-1}$$

where $\Lambda(\mathbf{x}-\mathbf{x}')$ is a kernel function; and Φ_a(x) is the kernel approximation of Φ(x). If $\Lambda(\mathbf{x}-\mathbf{x}')$ is the Dirac delta function, the kernel approximation will be able to exactly duplicate the original function. However, it is impossible to obtain a numerical solution using the Dirac delta function. Thus kernel functions that have finite support size are often used. The following kernel functions are often used in MLM [87]:

Cubic B-spline function

$$\Lambda(\mathbf{x}-\mathbf{x}') = \begin{cases} 2/3 - 4p^2 + 4p^3 & for\ \ 0 \le p \le 1/2 \\ 4(1-3p+3p^2-p^3)/3 & for\ \ 1/2 \le p \le 1 \\ 0 & for\ \ p \ge 1 \end{cases} \tag{2-2}$$

Exponential function

$$\Lambda(\mathbf{x}-\mathbf{x}') = \begin{cases} e^{-(p/a)^2} & for\ \ 0 \le p \le 1 \\ 0 & for\ \ p \ge 1 \end{cases} \tag{2-3}$$

and Quartic spline function

$$\Lambda(\mathbf{x}-\mathbf{x}') = \begin{cases} 1-6p^2+8p^3-3p^4 & for\ \ 0 \le p \le 1 \\ 0 & for\ \ p \ge 1 \end{cases} \tag{2-4}$$

where $p = \|(\mathbf{x}-\mathbf{x}')\|/d$; d is a design parameter that influences the effective region of the kernel function; and a is a constant.

The rationale of kernel estimation may be explained with an analogy. Suppose that we want to know the temperature of a given specific location $\mathbf{x}_0$ (x, y, z) in space. We measure it using a thermometer and obtain a temperature value. Due to the finite size of the thermometer, the value we obtained (in physical reality) is never the true temperature at the exact location of $\mathbf{x}_o$; rather, it is an approximation that integrates the effect of the surroundings. However our daily experience tells us that as long as the size of thermometer is much smaller than the change of the temperature in the space, this approximation is rather accurate. In kernel approximation, each kernel function can be imagined as a thermometer and the support size of the function is analogous to the size of the thermometer.

2.2.2 Reproducing Condition

For a finite domain, the kernel approximation in Equation (2-1) can be rewritten as:

$$\Phi(\mathbf{x}) = \int_{\Omega} \Lambda(\mathbf{x}-\mathbf{x}')\Phi(\mathbf{x}')d\mathbf{x}' \tag{2-5}$$

Equation (2-1) and (2-5) is the basic form of the Smoothed Particle Hydrodynamics (SPH) method which was proposed in the 1970s for studying the movement of stars. As one of

the ancient meshless methods, it is the inspiration for many other meshless methods. However this method fails to pass the convergence patch test; and the results obtained using this method have large numerical errors near the boundary region. Liu [119, 120] investigated the cause of the error and concluded that it is due to failure to meet the completeness or consistency conditions.

As is the case for FEM, the basis function must satisfy certain degrees of completeness conditions to ensure the convergence of the method. The degree of completeness is measured by the degree of polynomials that the basis function can exactly represent. For example, if an approximation with a basis function can exactly produce a zero-order polynomial, it is said that this basis function processes zero-order completeness. Zero-order completeness is the basic requirement for a MLM to converge.

In order to satisfy the completeness condition, the method is modified using the following procedure. Consider the following n^{th} order Taylor expansion of the original function.

$$\Phi(\mathbf{x}') = \Phi(\mathbf{x}) + (x'-x)\Phi^{'}(\mathbf{x}) + \frac{1}{2!}(x'-x)^2\Phi^{''}(\mathbf{x}) + \ldots + \frac{1}{n!}(x'-x)^n\Phi^{(n)}(\mathbf{x}) \tag{2-6}$$

Substituting Equation (2-6) into (2-5), we obtain

$$\Phi(\mathbf{x}) = m_0(x)\Phi(\mathbf{x}) + m_1(x)\Phi^{'}(\mathbf{x}) + m_2(x)\Phi^{''}(\mathbf{x}) + \ \ldots + m_n(x)\Phi^{(n)}(\mathbf{x}) \tag{2-7}$$

where

$$m_i(x) = \int_{\Omega} \Lambda(x-x')\frac{(x-x')^i}{n!}dx' \quad (i = 0,1,2,\ldots,n) \tag{2-8}$$

In order for Equation (2-7) to be valid, the kernel function must satisfy the following conditions:

$$m_0(x) = 1 \tag{2-9}$$

$$m_i(x) = 0 \quad (i = 1, 2, 3, ..., n) \tag{2-10}$$

The kernel functions given by Equation (2-2), (2-3) and (2-4), in general, will not satisfy the conditions (2-9) and (2-10). In order to satisfy these conditions, an enrichment function is added to the original kernel function such that

$$\Psi(\mathbf{x} - \mathbf{x}') = C(\mathbf{x}; \mathbf{x} - \mathbf{x}')\Lambda(\mathbf{x} - \mathbf{x}') \tag{2-11}$$

where $\Psi(\mathbf{x}\text{-}\mathbf{x}')$ is the modified kernel function; $C(\mathbf{x};\mathbf{x}\text{-}\mathbf{x}')$ is the enrichment function that varies with the location of approximation $\mathbf{x}$ and has the form of

$$C(\mathbf{x}; \mathbf{x} - \mathbf{x}_i) = \mathbf{h}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\mathbf{b}(\mathbf{x}) \tag{2-12}$$

where $\mathbf{h}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}_i) = \begin{bmatrix} 1 & (\mathbf{x}_1 - \mathbf{x}_{i1}) & \cdots & (\mathbf{x}_n - \mathbf{x}_{in})^n \end{bmatrix}$; $\mathbf{b}^{\mathrm{T}}(\mathbf{x}) = \begin{bmatrix} \mathbf{b}_0(\mathbf{x}) & \mathbf{b}_1(\mathbf{x}) & \cdots & \mathbf{b}_n(\mathbf{x}) \end{bmatrix}$.

Substituting the modified kernel function into Equation (2-8), we obtain

$$m_n(x) = \int_{\Omega} \mathbf{h}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\mathbf{b}(\mathbf{x})\Lambda(x - x')\frac{(x - x')^n}{n!}dx' \tag{2-13}$$

Reorganizing the above equation results in

$$m_n(x) = \left(\int_{\Omega} \mathbf{h}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\Lambda(x - x')\frac{(x - x')^n}{n!}dx' \right)\mathbf{b}(\mathbf{x}) \tag{2-14}$$

After substituting Equation (2-14) into Equations (2-9) and (2-10), we can write the reproducing condition into matrix form

$$\mathbf{P}(\mathbf{x})\mathbf{b}(\mathbf{x}) = \mathbf{h}(0) \tag{2-15}$$

where $\mathbf{h}^{\mathrm{T}}(0) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$; $\mathbf{P}(\mathbf{x})$ is a matrix whose elements are computed by

$$p_{ij}(\mathbf{x}) = \int_{\Omega} (\mathbf{x} - \mathbf{x}')^{i} (\mathbf{x} - \mathbf{x}')^{j} \Lambda(x - x') dx' \tag{2-16}$$

Finally, b(x) can be solved by inverting **P**.

$$\mathbf{b}(\mathbf{x}) = \mathbf{P}^{-1}(\mathbf{x})\mathbf{h}(0) \tag{2-17}$$

Substituting **b**(**x**) into (2-11), we have the expression for the modified kernel function

$$\Psi(\mathbf{x} - \mathbf{x}') = \mathbf{h}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\mathbf{P}^{-1}(\mathbf{x})\mathbf{h}(0)\Lambda(\mathbf{x} - \mathbf{x}') \tag{2-18}$$

2.2.4 Discretized Form of Kernel Approximation

In general, the integration in Equation (2-1) can not be solved analytically. Numerical integration is used such that the following discretized form can be obtained:

$$\Phi_a(\mathbf{x}) = \sum_{i=1}^{n} \Psi_i(\mathbf{x} - \mathbf{x}_i)\phi_i \tag{2-19}$$

where n is the number of nodes; $\mathbf{x_i}$ is the coordinate of i$^{\text{th}}$ node; ϕ_i is the nodal control value of i$^{\text{th}}$ node; and Ψ_{i} is the basis function associated with the i$^{\text{th}}$ node. The basis function has the following form:

$$\Psi_i(\mathbf{x}) = C(\mathbf{x}; \mathbf{x} - \mathbf{x}_i)\Lambda\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{d}\right) \tag{2-20}$$

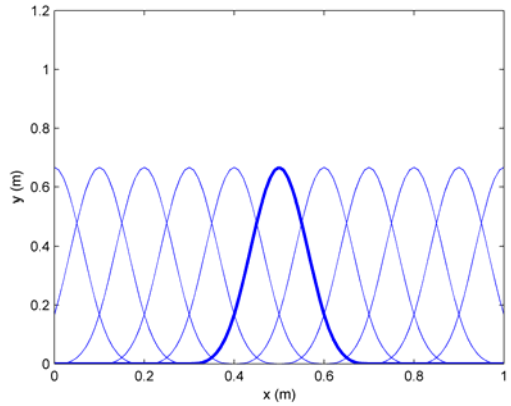
The discretized enrichment function $C(\mathbf{x}; \mathbf{x} - \mathbf{x}_i)$ is given by

$$C(\mathbf{x}; \mathbf{x} - \mathbf{x}_i) = \mathbf{h}^{\mathrm{T}}(0)\mathbf{P}^{-1}(\mathbf{x})\mathbf{h}(\mathbf{x} - \mathbf{x}_i) \tag{2-21}$$

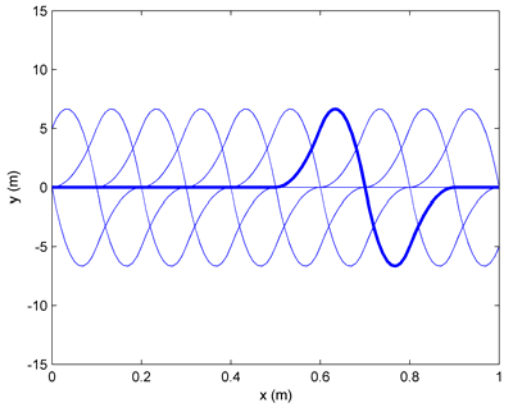
where $\mathbf{h}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_{\mathbf{i}}) = \begin{bmatrix} 1 & (\mathbf{x}_1 - \mathbf{x}_{i1}) & \cdots & (\mathbf{x}_n - \mathbf{x}_{in})^n \end{bmatrix}$; and $\mathbf{h}^{\mathrm{T}}(0) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$.

Unlike in FEM where ϕ_i is the nodal value or $\phi_i = \Phi(\mathbf{x}_i)$, $\Psi_i(\mathbf{x})$ may not be an interpolation function in the numerical formulation of MLM; that is, $\Psi_i(\mathbf{x}_i) \neq 1$ and $\Phi_i = \Phi(\mathbf{x}_i) \neq \phi_i$. To preserve the notation $\Phi_i = \Phi(\mathbf{x}_i)$, we call ϕ_i the nodal control value instead of the nodal value.

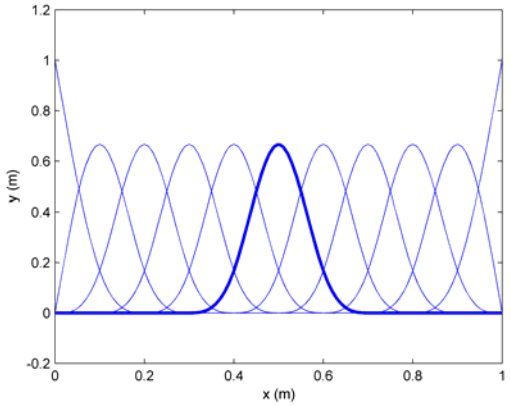
Figure 2-2(a) and (b) show the original cubic B spline functions and their derivative built on the computational domain $x \in [0,1]$ with 11 uniformly distributed nodes (at x=0, 0.1, 0.2, •••, 1). Figure 2-2(c) and (d) show the modified meshless basis functions and their derivatives. By comparing the new basis functions shown in Figure 2-2 with the original spline functions, it can be observed that the shapes of the meshless basis functions have been modified when they are transformed from the original B spline functions. Some significant differences can be observed for those located around the boundary. In addition, the new basis functions are no longer identical for all nodes like in the original B spline function. As stated earlier, such modifications are necessary for basis functions to satisfy consistent conditions that ensure the numerical approximation to converge.



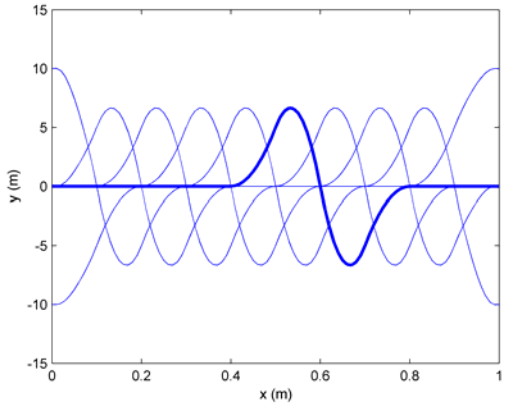
a) The original cubic B spline function



b) The derivative of the original cubic B spline function



c) Reproducing kernel basis function



d) The derivative of reproducing kernel basis function

Figure 2-2 Comparison of the original cubic spline and Meshless basis function

The meshless basis functions have following properties:

1. The support (non-zero region) of the modified kernel function is the same as the support of the original kernel function.

2. Although the original kernel function might be identical for all nodes, the modified kernel functions are, in general, different for different nodes. Furthermore they are

not an interpolating function, which means the value of the basis function is not 1 at its central nodal location.

3. According to the construction procedure of the meshless basis function, the final RKP basis function satisfies the nth order completeness condition. For n=0, the requirement we have from Equation (2-9) is

$$\sum_{i=1}^{n} \Psi_i(\mathbf{x} - \mathbf{x}_i) \equiv 1 \tag{2-22}$$

This property is called the partition unity property and is the basic requirement for a basis function to pass the theoretical convergence test. As will be shown in Chapter 4, this property is utilized in constructing the numerical integration for the adaptive computation of the meshless method.

4. The meshless basis function is more complicated and computationally involved than the FEM basis function.

2.2.5 Basis Function with Interpolating Properties

The meshless basis function, in general, does not have an interpolating property. As a result, it is not a trivial problem in the MLM weak-form formulation to apply an essential boundary condition as in FEM; additional computational steps are needed. The common methods for solving this problem include the Lagrange multiplier and the penalty method [9]. These methods require modifying the original weak-form formulation and thus, are difficult to use for solving such problems as the mechanical contact discussed in Chapter V.

In this research, we have utilized the method proposed in [121] to transfer a non-interpolating basis function into an interpolating basis function. The modified basis function $\hat{\Psi}_i(\mathbf{x})$ that can be computed from the RKP basis function (2-20) follows:

$$\hat{\Psi}_i(\mathbf{x}) = \sum_{j=1}^{n} \Psi_i(\mathbf{x}) L_{ij}^{-\mathrm{T}} \tag{2-23}$$

where the element L_{ij} is defined by

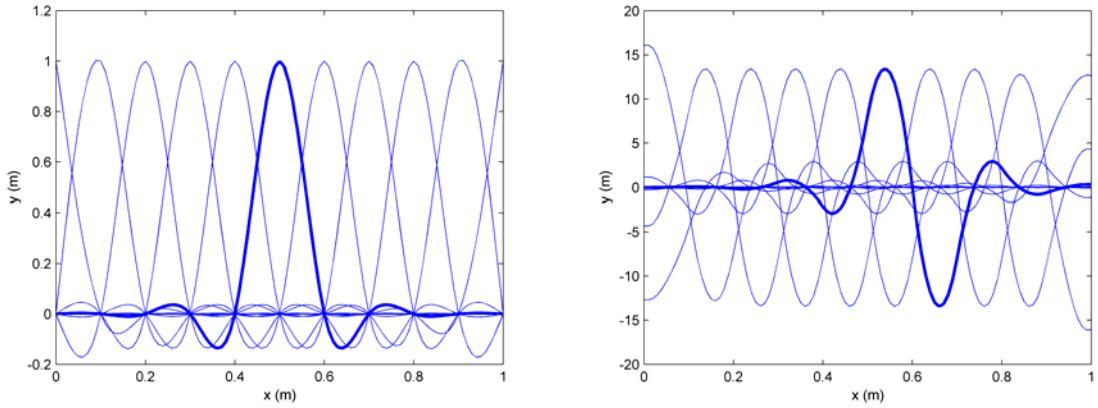
$$L_{ij} = \Psi_i(x_J) \tag{2-24}$$

The meshless approximation using a new interpolating basis function can be written as

$$\tilde{\Phi}(\mathbf{x}) = \sum_{i=1}^{n} \hat{\Psi}_i(\mathbf{x}) \Phi_i \tag{2-25}$$

where Φ_i now represents the value of the function at x_i and $\Phi_i = \Phi(x_i)$.

Figure 2-3 shows the transformed 1D reproducing kernel function. It can observed from Figure 2-3 that the transformed reproducing kernel functions become interpolating functions.



a) The RPKM basis function after transformation

b) The derivative of RPKM basis function after transformation

Figure 2-3 The reproducing basis function after transformation

2.4 Discretization Methods

In this section, we introduce two different methods of formulating MLM, a weak form method and a strong form method.

The weak form method is used extensively in FEM. It transforms the strong form governing equation into weak integral form such that the requirement for the differential order of the basis function is reduced. Although the solution obtained by this method in general does not exactly satisfy the strong form governing equation at the nodal locations, it can optimally minimize computational errors within whole the computational domain.

The other discretization method, the strong form method, has been commonly used in the finite difference method (FDM) for analyzing fluid mechanics. Unlike the Galerkin formulation, the strong form method obtains a system of discrete equations without integration. For linear problems, it is computationally simpler. Additionally,

more nodes can be easily appended to improve computational accuracy at desired local areas.

The following Poisson's problem is used to illustrate the formulations:

$$\mu_r \nabla^2 \Phi = f(\mathbf{x}) \tag{2-26}$$

where f is a function of the position vector $\mathbf{x}$; μ_r is a constant. Equation (2-26) is the governing equation for a number of physical problems such as heat transfer and electromagnetic field.

Two types of boundary condition are needed for solving the above problem. The first type of boundary condition, called the essential boundary condition, assigns the value of the field at the boundary according to a given function:

$$\Phi_\Gamma = g(\mathbf{x}) \tag{2-27}$$

where Φ_Γ is potential at the boundary; g($\mathbf{x}$) is a given function. Another type of boundary condition imposed upon the derivative of field is called the natural boundary condition, and it has the following form:

$$\nabla \Phi_\Gamma \bullet \mathbf{n} = h(\mathbf{x}) \tag{2-28}$$

where $\mathbf{n}$ is the normal at boundary; and $h(\mathbf{x})$ is a given function. This boundary condition is often specified at the interface at different materials, such as permanent magnet and air. How to handle such boundary condition will be discussed in Chapter 3.

2.4.1 Strong Form Discretization Method

The strong form formulation obtains the discretized system of equations by directly substituting the meshless approximation into governing equations and boundary

conditions. This method has some advantages. First, the strong form formulation is relatively easier to understand and implement. In addition, unlike the weak form formulation, this method does not need numerical integration such that the computational speed can be faster. Besides, it is easier to add nodes in this method since no background integration cells need to be reconstructed after nodal insertion. Although this method is also used in FEM, it is not popular primarily because the strong form formulation requires the basis function to be higher-order differentialable at nodal locations while the FEM basis function is often non differentialable or only first-order differentialable at these locations.

The process begins with dividing the computational domain into individual nodes. Based on the different location in the computational domain, the nodes can be broadly classified into three types; the nodes located inside the boundary, the nodes located at the boundary specified with essential boundary condition, the node located at the boundary specified with natural boundary condition. At the location of the j^{th} node inside the boundary, substituting Equation (2-19) into Equation (2-26) yields

$$\mu_r \sum_{i=1}^{n} \nabla^2 \Psi_i(\mathbf{x}_j) d_i = f(\mathbf{x}_j) \tag{2-29}$$

where i represents i^{th} basis function; n is the total number of nodes.

Similarly, for the nodes on the boundary characterized by the essential boundary condition and by the natural boundary condition, applying Equation (2-19) to Equation (2-27) and (2-28) accordingly gives two equations respectively:

Natural B.C. $$\sum_{i=1}^{n} \Psi_i(\mathbf{x}_j) d_i = g(\mathbf{x}_j) \tag{2-30}$$

Essential B.C. $$\left(\sum_{i=1}^{n} \nabla \Psi_i(\mathbf{x}_j) d_i \right) \bullet \mathbf{n}_j = h(\mathbf{x}_j) \tag{2-31}$$

Since each node results in one discretized equation, this procedure eventually leads to a $n \times n$ linear system of equations which can be solved using methods such as Gauss-Seidel method.

2.4.2 Weak Form Discretization Methods

Unlike SFF where the essential boundary condition is satisfied by directly substituting meshless approximation into the boundary condition, in WFF, the governing equation is transformed into an integral form so that the essential boundary condition can only be applied after its discretized form is obtained. The RPK basis function given by Equation (2-20), in general, does not have the interpolation property and the modified RPK basis function Equation (2-23) is used instead for the WFF.

The weak form equations are derived using the Galerkin method: First, the meshless approximation Equation (2-25) is substituted into the governing equation. Both sides of the equation are then multiplied by a test function. In WFF, both the test and trial functions are from the same functional space. Next, the resulting equation is integrated over the entire domain. For Equation (2-29), we have

$$\oint_{\Omega} \Psi_j(\mathbf{x}) \mu_r \nabla^2 \left(\sum_{i=1}^{n} \Psi_i(\mathbf{x}) \Phi_i \right) d\Omega = \oint_{\Omega} \Psi_j(\mathbf{x}) f(\mathbf{x}) d\Omega \tag{2-32}$$

The left side of the above equation can be integrated by parts:

$$\oint_{\Omega} \mu_r \nabla \left(\Psi_j(\mathbf{x}) \nabla \sum_{i=1}^{n} \Psi_i(\mathbf{x}) \Phi_i \right) d\Omega - \oint_{\Omega} \mu_r \nabla \Psi_j(\mathbf{x}) \nabla \left(\sum_{i=1}^{n} \Psi_i(\mathbf{x}) \Phi_i \right) d\Omega \tag{2-33}$$

The 1st term in Equation (2-33) is transformed from volume integral to surface integral using the divergence theorem:

$$\oint_{\Gamma} \mu_r \Psi_j(\mathbf{x}) \nabla \left(\sum_{i=1}^{n} \Psi_i(\mathbf{x}) \Phi_i \right) \bullet \mathbf{n} d\Gamma \tag{2-34}$$

Applying the natural boundary condition given by Equation (2-28) to yield

$$\oint_{\Gamma} \mu_r \Psi_j(\mathbf{x}) \nabla \left(\sum_{i=1}^{n} \Psi_i(\mathbf{x}) \Phi_i \right) \bullet \mathbf{n} d\Gamma = \oint_{\Gamma} \mu_r \Psi_j(\mathbf{x}) h(\mathbf{x}) d\Gamma \tag{2-35}$$

Thus, Equation (2-32) can then be rewritten as

$$\oint_{\Omega} \mu_r \nabla \Psi_j(\mathbf{x}) \nabla \left(\sum_{i=1}^{n} \Psi_i(\mathbf{x}) \Phi_i \right) d\Omega = \oint_{\Gamma} \mu_r \Psi_j(\mathbf{x}) h(\mathbf{x}) d\Gamma - \oint_{\Omega} \Psi_j(\mathbf{x}) f(\mathbf{x}) d\Omega \tag{2-36}$$

The numerical integration methods, such as Gaussian quadrature, have to be applied for Equation (2-36) to obtain the discretized system of equations. Finally the essential boundary condition has to be imposed to the discretized equations.

2.5 Numerical Examples

The following Poisson problem is chosen as an example to compare the two discretization methods:

$$\nabla^2 u = -8\pi^2 \cos(2\pi x) \cos(2\pi y) \qquad 0 < x, y < 1 \tag{2-37}$$

with the following essential boundary conditions:

$$u = \cos(2\pi y) \quad \text{at} \quad x = 0, 1$$

and $\quad u = \cos(2\pi x) \quad \text{at} \quad y = 0, 1$

Figure 2-4 shows the equal potential plot of the problem. The exact solution of this problem is given by

$$u = \cos(2\pi x)\cos(2\pi y) \tag{2-38}$$

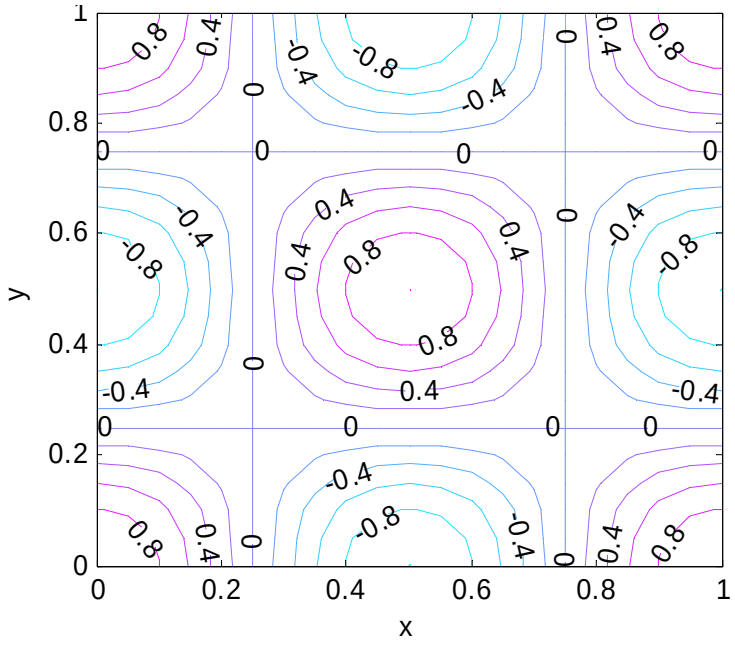


Figure 2-4 Equal potential plot

For the purpose of validating the computational algorithm, we compare the computational result against the exact solution. For this reason, we define the computation error as follows:

$$error = \sqrt{\iint (u_{computed} - u_{exact})^2 \, d\Omega} \tag{2-39}$$

We also evaluate the computational efficiency of the two MLM's using FEM for benchmark comparison. While the basis function in MLM is highly nonlinear and not in

closed form, the FEM basis function is constructed based on the meshes. The FEM basis functions are normally piecewise polynomial that interpolate the approximated function among the element nodes.

In this comparison, the following rectangular 2D bilinear element has been used for the FEM as a basis for comparison:

$$\Psi_i(x,y) = c_0 + c_1 x + c_2 y + c_3 xy \tag{2-40}$$

In general, the coefficient of polynomial can be obtained by imposing the interpolating condition, and then the element equation can be computed. As the element equation can be more efficiently computed using the natural coordinate system, the following standard form of an iso-parametric shape function is a as follows.

$$\Psi_i(\xi,\eta) = (1+\xi\xi_i)(1+\eta\eta_i)/4 \tag{2-41}$$

where ξ and η are a natural coordinate system; and ξ_i and η_i are the coordinate of the i^{th} node. Thus, the corresponding weak form integration has the following form:

$$\int_{\Omega} f(x,y)d\Omega = \int_{-1}^{1}\int_{-1}^{1} f(x(\xi,\eta),y(\xi,\eta))J(\xi,\eta)d\xi d\eta \tag{2-42}$$

where
$$J(\xi,\eta) = \det\begin{vmatrix} \partial x/\partial\xi & \partial x/\partial\eta \\ \partial y/\partial\xi & \partial y/\partial\eta \end{vmatrix} \tag{2-43}$$

The numerical integration for the FEM is the 2×2-point Gaussian integration routine. For the MLM-WFF, the 4×4-point Gaussian integration is chosen because of its highly nonlinear basis function. The errors are compared in Table 2-1 and Figure 2-5 where the nodal distance refers to the spacing between two adjacent nodes. In Table 2-1, the convergence rate is the average slope in Figure 2-5.

Table 2-1: Comparison of computation error (time in seconds)

Nodes	MLM-WFF*	MLM-SFF	Linear FEM^{+}
5×5	7.7E-2 (0.37s)	1.47E-1 (0.22s)	1.72E-1(0.059s)
9×9	2.1E-3 (2.7s)	1.39E-2 (0.65s)	4.5E-2(0.069s)
17×17	1.57E-4 (28.3s)	1.5E-3 (2.7s)	1.13E-2(0.157s)
Convergence rate	2.97	2.90	1.98

* 4×4-point Gaussian integration; + 2×2-point Gaussian integration.

The following are some observations from results:

1. As shown in Figure 2-5, all three methods converge to the exact solution given a sufficient number of nodes. Both MLM's (that use higher order shape function than FEM) achieve a higher converging rate than the linear FEM.

2. Both MLM's use the same higher-order shape function, which means their solutions are in the same functional space. However, the Galerkin formulation in MLM-WFF ensures that its solution is optimal in the same functional space while the SFF (the shape function does not have an interpolating property) only satisfies the governing equation at the nodes. The results due to this difference can be verified with the error data in Table 2-1. For the 17× 17 nodes, the accuracy of MLM-WFF is an order higher than that of MLM-SFF and two-orders higher than that of FEM.

3. Due to the highly nonlinear shape function, the computation time of MLM's is significantly higher than FEM. The need for higher order numerical integration also significantly increases the computation time of MLM-WFF. A numerically more efficient integration scheme suitable for MLM-WFF is needed.

This example shows that MLM-SFF represents an interesting trade-off between MLM-WFF and FEM. Domain discretization in MLM-SFF is simple, and requires no meshes and no integration to derive the numerical model.

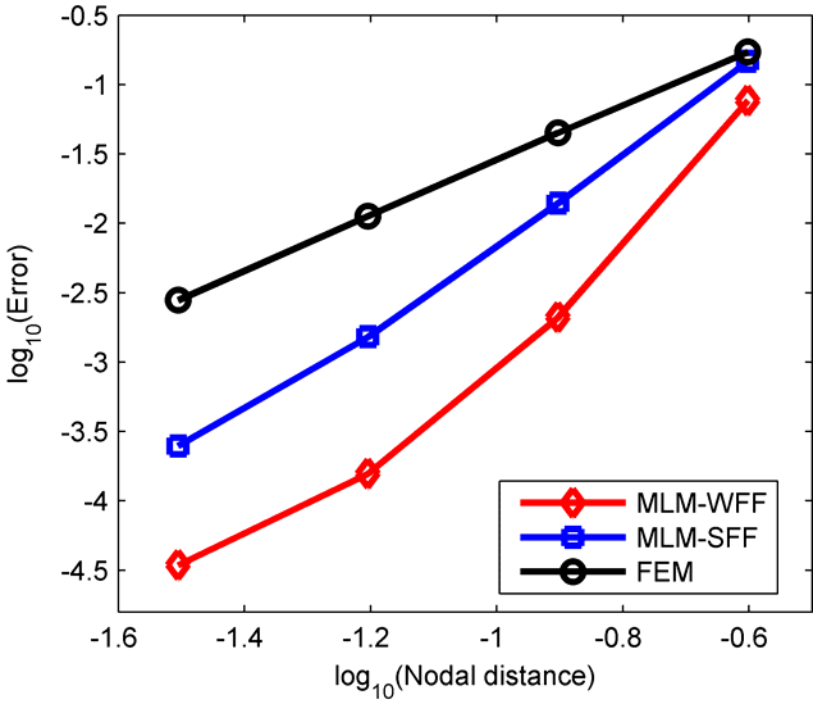


Figure 2-5 Rate of convergence

2.4 Summary

In this chapter, the reproducing kernel particle method for constructing a meshless basis function and two methods for formulating MLM are reviewed. The convergence speed and computational time of the two discretization formulations are investigated and compared against the solution computed using FEM. One of difficulties in the meshless method is to apply the essential boundary conditions. The original basis function constructed using the reproducing kernel technique does not possess the interpolating property. However, a linear transformation (which can be computed during the pre-

process) can be applied to transform it into an interpolating basis function. The basis function introduced here is used throughout the rest of the dissertation.

CHAPTER III

HANDLING DISCONTINUITY IN MESHLESS METHOD

3.1 Overview

This chapter discusses the methods of handling discontinuity in the computational domain when computing with MLM. As an illustration, we consider the problem of magnetic force computation, which is an important step to design a magnetic actuator. Methods to compute magnetic force relied on the solution of the magnetic field. In practice, the computational accuracy for the magnetic field around the boundary of a permanent magnet is very important since the air gap between permanent magnets is often chosen to be very small for increasing the magnetic force. Despite its importance, a high-accuracy solution for this region is difficult to obtain in MLM because the magnetic flux intensity is discontinuous at the boundary of permanent magnet but the ML basis function is a smooth function in general. Thus special treatments must be applied to approximate the discontinuity of the field intensity while maintaining the zero order continuity of the potential.

The remainder of this chapter is organized as follows:

1. We briefly introduce the governing equations for the magnetic field problem and the formulation of magnetic field scalar potential.
2. Next, the methods for handling the discontinuity in the magnetic field are developed for two discretization methods; the strong form formulation, and the weak form formulation.

3. This is followed by numerical examples where the exact solution is available to validate the computation of both MLM's. This example also provides a means to examine the effects of MLM's handling of the discontinuity at the material interface on the computation of magnetic fields.

4. Finally, we compute in the second example the magnetic forces between two permanent magnets. This example offers an alternative means to validate the MLM by comparing its computational results against those measured experimentally.

3.2 Scale Potential Formulation of Magnetic Field Computation

3.2.1 Governing Equations

The field of a magnetic system is governed by a set of Maxwell's equations:

$$\nabla \bullet \mathbf{B} = 0 \tag{3-1}$$

$$\nabla \times \mathbf{H} = \mathbf{J} \tag{3-2}$$

where **H** is the magnetic field intensity; **B** is the flux density; and **J** is the current density.

In addition, the flux density **B** and magnetic field intensity **H** are related by the constitutive equation:

$$\mathbf{B} = \mu_0 \mu_r \mathbf{H} + \mu_0 \mathbf{M} \tag{3-3}$$

where **M** is the magnetization; μ_0 is the permeability of the free space; μ_r is the relative permeability of the material.

Two methods, vector potential formulation and reduced scalar potential formulation, are often used to reduce Equations (3-1) and (3-2) to a more compact and

solvable form. Vector potential formulation has a long history being utilized in FEM for solving 2D electro-magnetic field problems. However, the computational time for this method increases significantly when it is extended to 3D because the number of variables needed for each node has been increased from one to three in this case. Unlike the vector potential formulation, the scale potential formulation only used one variable for each node no matter if the problem was 2D or 3D. Thus, we choose to use the scalar potential formulation in order to reduce the effort for the future development.

To obtain the reduced potential, the magnetic field intensity **H** is first divided into two parts: the field induced by the electrical current denoted as $\mathbf{H}_s$ and the field induced by permanent magnetic material denoted as $\mathbf{H}_m$.

$$\mathbf{H} = \mathbf{H}_s + \mathbf{H}_m \tag{3-4}$$

By doing this, Equation (3-2) is separated into

$$\nabla \times \mathbf{H}_s = \mathbf{J} \tag{3-5}$$

and

$$\nabla \times \mathbf{H}_m = 0 \tag{3-6}$$

For a source free problem, the current source is neglected such that $\mathbf{H}_s \equiv 0$. To simplify the notation, we refer $\mathbf{H}_m$ as **H** for the rest of this dissertation.

Equation (3-6) implies that the magnetic field induced by the magnetic material $\mathbf{H}_m$ is an irrotational field. The fact that the irrotational field can be mathematically described by a scale field leads to the following definition:

$$\mathbf{H} = -\nabla\Phi \tag{3-7}$$

where Φ is the magnetic scalar potential. By substituting Equation (3-7) into Equation (3-3), the magnetic flux density can be expressed in terms of the scalar potential as

$$\mathbf{B} = -\mu_0\mu_r\nabla\Phi + \mu_0\mathbf{M} \tag{3-8}$$

Finally, we obtain the governing equation by substituting **B** from Equation (3-8) into Equation (3-1):

$$\nabla\bullet\left(\mu_0(-\mu_r\nabla\Phi)\right) + \nabla(\mu_0\mathbf{M}) = 0 \tag{3-9}$$

It is easy to see that Equation (3-9) matches the general form of Poisson's problem given in Equation (2-26).

In general, **M** is a nonlinear function of **H**. However, for a system containing permanent magnets and air (which are two materials used extensively in the subsequent computation), the magnetization can be approximated as a constant depending on the material [122]. In this case, the governing Equation (3-9) can be written as

$$\nabla(\mu_0\nabla\Phi) = \nabla(\mu_0\mathbf{M}_0) \tag{3-10}$$

where $\mathbf{M}_0$ is the magnetization. Since $\mathbf{M}_0$ is constant, $\nabla(\mu_0\mathbf{M}_0) \equiv 0$ such that Equation (3-10) is reduced to

$$\nabla(\mu_0\nabla\Phi) = 0 \tag{3-11}$$

3.2.2 Boundary Conditions (BC's)

In a magnetic field, the boundary condition at the infinitely far boundary can be expressed as:

$$\Phi_{\mathbf{x}\to\infty} = 0 \tag{3-12}$$

In addition, at the interface between two different materials (denoted as regions *p* and *q*), the following BC's must be satisfied:

1. **H** is continuous along the tangential direction at the material interface, or

$$(\mathbf{H}_p - \mathbf{H}_q) \times \mathbf{n} = 0 \tag{3-13}$$

2. **B** is continuous along the normal of the interface:

$$\mathbf{B}_p \bullet \mathbf{n} = \mathbf{B}_q \bullet \mathbf{n} \tag{3-14}$$

In terms of scalar potential functions, (3-13) and (3-14) corresponds to

$$\Phi_p = \Phi_q \tag{3-15}$$

$$\left(\mu_{rq}\nabla\Phi_q - \mu_{rp}\nabla\Phi_p\right)\bullet\mathbf{n} = \left(\mathbf{M}_q - \mathbf{M}_p\right)\bullet\mathbf{n} \tag{3-16}$$

3.2.3 Magnetic Force Computation

In this research, the Maxwell stress tensor and the Lorenz force law are used to compute the magnetic force between two permanent magnets and the magnetic force between permanent magnets and electrical coils respectively.

Maxwell stress tensor

The magnetic force **F** between magnets is obtained by integrating the Maxwell stress tensor:

$$\mathbf{F} = \oint_{\Gamma} \mathbf{T}\bullet\mathbf{n}d\Gamma \tag{3-17}$$

where $$\mathbf{T} = [\mathrm{T}]\bullet\mathbf{n} = \left(\mathbf{B}(\mathbf{B}\bullet\mathbf{n}) - \frac{1}{2}B^2\mathbf{n}\right); \tag{3-18}$$

F is the magnetic force; Γ is an arbitrary boundary enclosing the body of interest; and **n** is the normal of the material interface.

Lorenz force law

Lorenz force law states that magnetic force can be obtained by integrating the force density exerted on the current carrying conductor by its interaction with the magnetic field:

$$\mathbf{F} = \int_V (\mathbf{J} \times \mathbf{B}) r dr d\theta dz \tag{3-19}$$

where V is the volume of the current conductor.

3.3 Handling Magnetic Discontinuity

Equation (3-16) implies that discontinuity could happen for the magnetic field intensity along the normal direction of material interface. In [123], a method based on The Lagrange multiplier is proposed for weak form formulation to solve the discontinuity problem. However, the Lagrange multiplier method can only approximately satisfy the continuity constraint of the magnetic field intensity along the tangential direction at the interface. In [9], a method based on partition of unity theory was proposed for solving the discontinuity problem in a mechanical system.

An alternative to eliminate the difficulty of handling boundary conditions at the interface and to avoid numerical integration in deriving the discretized system equations is to use the strong form of the governing equation. As shown in Chapter 2, the strong

form formulation is more flexible than the weak form formulation in terms of handling the boundary condition.

3.3.1 The Method for Strong Form Formulation (SFF)

In SFF (or the point collocation method), the problem domain is divided into subregions based on their material properties and discretized into n nodes consisting of n_ℓ interior, n_f far-field boundary and n_m material interface nodes. To solve for the n nodal control values, we derive the equations by substituting the approximate solution (9) into the governing Equation (3-11) and the BC's (3-12), (3-15) and (3-16).

Example 3.1: Illustration of node assignment

As an illustration, consider Figure 3-1 with two regions (k=1 and 2, where region 1 is a free air space). The interior, material boundary and far field nodes are represented by circles, squares and triangles respectively. Each of the points at the material boundary is shared by two regions and represented by two overlapping square nodes. The number and type of nodes are listed in Table 1 where n_k is the number of nodes in the k^{th} region; and the number of nodal control values to be solved in this example is 52.

Table 3-1: Node number for the example shown in Figure 3-1

Region/Nodes	Interior	Far field	Interface, n_m	n_k
k=1	12	12	12	36
k=2	4	0	12	16
Subtotal	n_ℓ =16	n_f=12	$2n_m$ =24	$n+ n_m$ =52

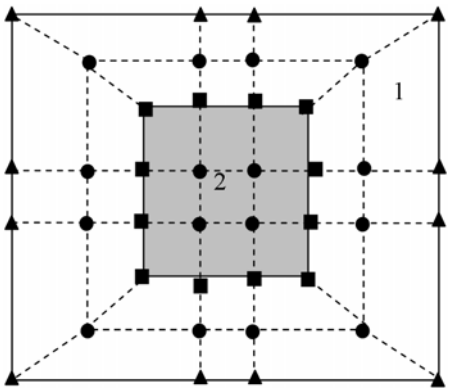


Figure 3-1 Example domain with two different materials

The system of equations for the general case can be obtained as follows. For the interior node, the substitution of Equation (2-19) into Equation (3-11) at the j^{th} node of the k^{th} region yields

$$\mu_k \sum_{i=1}^{n_k} \nabla^2 \Psi_{ki}(\mathbf{x}_{kj}) d_{ki} = f_k(\mathbf{x}_j) \tag{3-20}$$

where μ_k is the relative permeability of the material in the k^{th} region. Similarly, applying Equation (2-19) to the j^{th} node at the far field Equation (3-12) gives one of the n_f boundary node equations.

$$\sum_{i=1}^{n_1} \Psi_{fi}(\mathbf{x}_{fj}) d_{fi} = 0 \tag{3-21}$$

where the subscript f denotes the far-field region. The two overlapping nodal control values (which share the same coordinate at the interface) must satisfy Equation (3-15) and Equation (3-16). Using Equation (2-19), Equation (3-15) becomes

$$\sum_{i=1}^{n_1} \Psi_{pi}(\mathbf{x}_{pj}) d_{pi} - \sum_{i=1}^{n_2} \Psi_{qi}(\mathbf{x}_{qj}) d_{qi} = 0 \tag{3-22}$$

where the subscripts "p" and "q" denote the two overlapping nodes in p^{th} and q^{th} regions; and $\mathbf{x}_{pj} = \mathbf{x}_{qj}$. Implementation of condition Equation (3-16) is divided into the following two cases:

Case A: Boundary with a well-defined normal at the node

For a smooth boundary where the normal of the boundary at the nodal location exists, Equation (3-16) can be written as

$$\left(\mu_{r2} \sum_{j=1}^{n_2} \nabla \Psi_{2j}(\mathbf{x}_{2p}) d_{2j} - \mu_{r1} \sum_{i=1}^{n_1} \nabla \Psi_{1i}(\mathbf{x}_{1q}) d_{1i} \right) \bullet \mathbf{n} = \left(\mathbf{M}_2 - \mathbf{M}_1 \right) \bullet \mathbf{n} \tag{3-23}$$

Case B: Corner node where the normal does not exist

In general, Equation (3-23) is invalid at the corner where the normal does not exist. Although an average using both sides of the corner is often used to approximate the normal at the corner, this method (though consistent with the formulation) infers significant errors around the corner. Instead, we use the Gauss integral law of the flux density given in Equation (2-24).

$$\oint_{\Gamma} \mathbf{B} dA = 0 \tag{3-24}$$

As illustrated in Figure 3-2, a small square virtual boundary (with size *h)* is setup at the corner; $\underline{\mathbf{x}}_1$, $\underline{\mathbf{x}}_2$, $\underline{\mathbf{x}}_3$ and $\underline{\mathbf{x}}_4$ are the corner coordinates of the virtual square; s_1, s_2, s_3 and s_4 are the virtual surfaces; and $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$ and $\mathbf{n}_4$ are their respective normal.

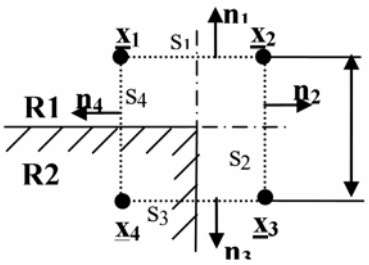


Figure 3-2 Material boundary condition at a corner

The flux passing though the surface s_1 can be approximated by $h[(\mathbf{B}_1+\mathbf{B}_2)/2]\bullet\mathbf{n}_1$ and similarly the fluxes passing through other surfaces can be calculated accordingly. The total fluxes passing though the boundary of the virtual square must be zero, which leads to

$$\sum_{i=1}^{4}\left[\left(\mathbf{B}_i+\mathbf{B}_{i+1}\right)\bullet\mathbf{n}_i\right]=0 \tag{3-25}$$

where i=5 is essentially repeating i=1; from Equation (3-3) and Equation (2-19)

$$\mathbf{B}_i=\mu_{r\ell}\sum_{k=1}^{n_\ell}\Psi_{\ell j}(\underline{\mathbf{x}}_i-\mathbf{x}_k)d_{\ell j}+\mathbf{M}_\ell,\ \ell=\begin{cases}1 & \text{if } i=1,2,3\\ 2 & \text{if } i=4\end{cases}$$

and where ℓ and n_ℓ denotes the region and its number of nodes respectively. Equation (3-25) is used to satisfy condition Equation (3-16) at the corner where the normal is not well defined.

The above procedure can be applied to the remaining nodes resulting in n_ℓ interior, n_f far-field and n_m boundary interface nodes in the forms given by Equations (3-20), (3-21), (3-22), (3-23) and (3-25) respectively. Once the problem domain is discretized, the approximate solution can be solved from Equations (3-20), (3-21), (3-22), (3-23) and (3-25) with the shape function given by Equation (2-20).

3.3.2 The Method for Weak Form Formulation (WFF)

In this section, we first introduce the method used in WFF for handling the material boundary condition. Then we present the method to improve the approximation at material boundary in meshless WFF.

Apply material boundary condition in Galerkin method

The application of the BC's can be illustrated using Figure 3-1. Unlike SFF where two nodal control values are used for a coordinate point at the material interface, the WFF uses a single shape function to approximate the problem domain as a whole. Thus, the continuity of potential at material boundary Equation (3-15) is satisfied automatically. The continuity of the magnetic flux density along the normal of the material interface Equation (3-16) is integrated into the weak-form. Apply the boundary condition Equation (3-16), the first term of Equation (2-36) becomes

$$\oint_{\Gamma} \mu_r \Psi_j(\mathbf{x}) \nabla \left(\sum_{i=1}^{n} \Psi_i(\mathbf{x}) \Phi_i \right) \bullet \mathbf{n} d\Gamma = \oint_{\Gamma_m} \Psi_j (\mathbf{M}_1 - \mathbf{M}_2) \bullet \mathbf{n} d\Gamma \tag{3-26}$$

where Γ_m denotes the material boundary. Equation (2-36) becomes

$$\oint_{\Gamma_m} \Psi_j (\mathbf{M}_1 - \mathbf{M}_2) \bullet \mathbf{n} d\Gamma - \oint_{\Omega} \mu_r \nabla \Psi_j(\mathbf{x}) \nabla \left(\sum_{i=1}^{n} \Psi_i(\mathbf{x}) \Phi_i \right) d\Omega = \oint_{\Omega} \Psi_j(\mathbf{x}) f(\mathbf{x}) d\Omega \tag{3-27}$$

The continuity of the flux density **B** along the normal of the material interface, Equation (3-16), is ensured by the WFF Equation (3-27). Once the global stiffness matrix is formed, the Dirichlet boundary condition Equation (3-11) at far field can be applied by using the interpolating shape function Equation (2-23).

Approximate the discontinuity at the material interface in weak form formulation

BC Equations (3-13) and (3-14) imply that **H** is discontinuous along the normal of the material interface. However, the RPK basis functions are, in general, continuous. Without dividing the computational domain into sub regions in MLM-WFF, the continuous meshless basis functions often result in numerical errors in the region near the material interface. To solve this problem, a method based on partition of unity theory [9] that was proposed for a discontinuity problem in mechanical systems is adapted here for the computation of the magnetic field. For this, we modify the approximate solution Equation (2-23) near the boundary (Figure 3-3) as follows:

$$\tilde{\Phi}(x) = \sum_{i=1}^{n} \hat{\Psi}_i(x)\Phi_i + \sum_{i=1}^{n_d} N_i(s)\Theta(r)b_i \tag{3-28}$$

where n_d is the number of nodes at material interface; N_i is a one-dimensional shape function of the arc length s along the discontinuity interface between two adjacent nodes; r is the distance along the normal to the discontinuity interface between the two nodes; and b_i is the strength of the discontinuity to be solved. In Equation (3-28), Θ is a function with a discontinuous first-order derivative at $\boldsymbol{x}_a$, the location of the discontinuity. Equation (3-29) shows an example of a discontinuous function:

$$\Theta(r) = \begin{cases} (-r^3 + 3r^2 - 3r + 1)/6 & r \le 1 \\ 0 & r \ge 1 \end{cases} \tag{3-29}$$

where $r = \left|\mathbf{x} - \mathbf{x}_a\right| / r_{\max}$. Thus, the number of unknown to be solved in the WFF is $n+n_d$.

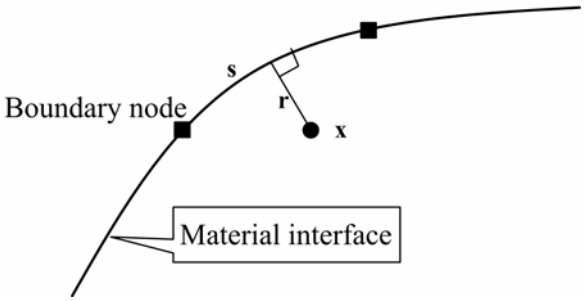


Figure 3-3 Discontinuity function at material boundary

3.4 Computational Results and Discussions

A MATLAB program was written for computing the magnetic field. Numerical results were obtained for three examples:

Example 3.2: This example illustrates the effect of boundary conditions at the material interface on the computation.

Example 3.3: We then compute the magnetic forces between two permanent magnets.

Example 3.4: In this example, the computed magnetic forces are compared against published experimental data conducted at Georgia Tech [124].

In the following comparison, we use Equations (2-23), (2-24), (2-25) in addition to Equations (2-19), (2-20), (2-21) to impose the boundary condition in the MLM-WFF while the basis function for MLM-SFF retains the original form Equations (2-19), (2-20), (2-21).

Example 3.2

We use the two MLM's to solve for the magnetic field intensity around a cylindrical permanent magnet in free space as shown in Figure 3-4. The interest here is to

investigate the effect of the numerical approximation on the magnetic field around the material interface. The cylindrical magnet shown in Figure 3-4 is uniformly magnetized in the z direction, $\mathbf{M} = M_o \mathbf{i}_z$, which **M** has no divergence and thus $f(\mathbf{x}) = 0$ throughout the volume. Thus, the source of **H** is on the surfaces where **M** originates and terminates. A closed-form solution of the potential and magnetic field intensity along the z axis, H_z, can be derived using the superposition-of-integral [89] of magnetic charges **c**.

The general solution of Equation (3-9) with μ_r=1 is as follows:

$$\Phi = \int_{\Omega} \frac{f(\mathbf{c})}{4\pi\mu_0 |\mathbf{x} - \mathbf{c}|} d\Omega \tag{3-30}$$

where $f(\mathbf{c}) = \nabla \bullet \mathbf{M}$. For a uniformly magnetized cylinder shown in Figure 3-4, the potential at the location **c**(x, y, z) is

$$\Phi(\mathbf{s}) = \frac{M_0}{4\pi} \int_0^{2\pi} \int_0^R r\left(\frac{1}{\Delta_-} - \frac{1}{\Delta_+}\right) dr d\theta \tag{3-31}$$

where $$r = \sqrt{x^2 + y^2}$$

and $$\Delta_{\mp} = \sqrt{\left(z \mp \frac{d}{2}\right)^2 + \left(y \mp r\sin\theta\right)^2 + \left(r\cos\theta\right)^2} \tag{3-32}$$

The Closed-form solution of the potential and magnetic intensity field along the z axis is given by:

$$\frac{\Phi_z}{dM_0} = \frac{1}{2}\left[\left(A_- - |B_-|\right) - \left(A_+ - |B_+|\right)\right] \tag{3-33}$$

and

$$\frac{H_z}{M_0} = \frac{1}{2}\left[\frac{B_-}{A_-} - \frac{B_+}{A_+} + \gamma\right] \tag{3-34}$$

where

$$A_{\mp} = \sqrt{\left(\frac{R}{d}\right)^2 + \left(\frac{z}{d} \mp \frac{1}{2}\right)^2}\ ; \tag{3-35}$$

$$B_{\mp} = \frac{z}{d} \mp \frac{1}{2}; \tag{3-36}$$

$$\gamma = \begin{cases} 0 & \text{when } |z| \geq \dfrac{d}{2} \\ 2 & \text{when } |z| < \dfrac{d}{2} \end{cases}$$

The magnetic field intensity has a discontinuity at its circumferential material interface ($z=0.5$). We compare the computational results of the MLM-WFF and MLM-SFF against the closed-form solutions Equation (3-33) and Equation (3-34).

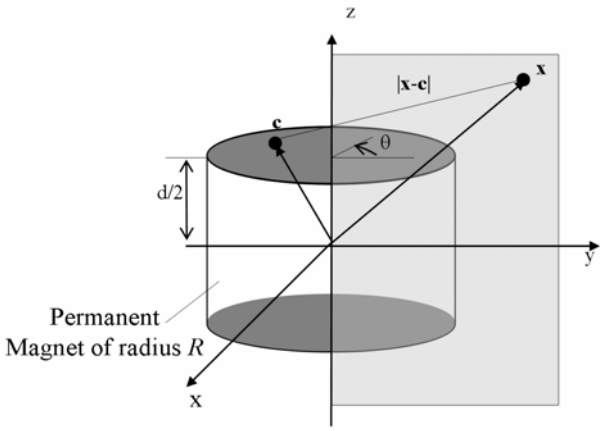


Figure 3-4 Cylindrical magnet is free air space

Both ML methods are formulated in cylindrical coordinates.

$$\nabla^2\Phi = \frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\Phi}{\partial\rho}\right)+\frac{1}{\rho^2}\frac{\partial^2\Phi}{\partial\theta^2}+\frac{\partial^2\Phi}{\partial Z^2}=0 \tag{3-37}$$

where $\rho=r/R$, $Z=z/R$; and R is the radius of the magnets.

For this axi-symmetric problem, we have

$$\frac{\partial^2\Phi}{\partial\theta^2}=0 \tag{3-38}$$

In addition to the boundary condition Equations (3-15) and (3-16), we have the following boundary condition for the symmetry:

$$\frac{\partial\Phi}{\partial r}=0 \quad \text{at } r=0 \tag{3-39}$$

Table 3-2: Parameters used in Example 3.2

Parameters	Value
Node number in MLM -WFF (in the *y*-*z*)	41×41
Node number in MLM –SFF (in the *y*-*z*)	151×201
Magnetic Radius/Magnetic thickness *R/d*	1
Magnetization M_0 (A/m)	1

The simulation parameters are listed in Table 3-2. The computed Φ and H_z are compared in Figure 3-5 and Figure 3-6 respectively. Both ML methods give a reasonable prediction of the potential and magnetic field intensity. As shown in Figure 3-6(a), MLM-WFF has a very good overall prediction but exhibits some oscillations near the discontinuous interface. The discontinuity function Equation (3-29) added to the MLM-WFF improves the accuracy of the prediction near the interface but can not completely remove the oscillations. SFF provides smoother field intensity than WFF, as it solves the

entire domain by dividing it into sub-domains and has no discontinuity within each sub-domain. Figure 3-6(b) shows the effect of the corner where the normal does not exist on the computation of the field intensity using MLM-SFF. Since the closed form solution is only valid along the z-axis, we compute the analytical solution by numerical double integration of (3-31). As shown in Figure 3-6(b), the computation using the Gauss integral offers a reasonable estimate as compared to that calculated using the average normal.

Example 3.3

Poisson's Equation is linear; thus, the principle of superposition applies and is used to compute the repulsive force between two identical permanent magnets separated by an air-gap *g*. The overall intensity can be obtained from the algebraic sum of the two permanent magnets individually. Once **H** (and hence **B)** is known, the overall force on a body can be computed using surface integration of the Maxwell stress tensor from Equation (3-17).

Once the magnetic field on the *y-z* plane is computed (see Example 3.2), the magnetic field at any general location **x**(x, y, z) can be computed from a corresponding pre-computed point **x'**(*0, y', z'*) on the *y-z* plane using the property of axi-symmetry as illustrated in Figure 3-7. This can be done as follows. Given the coordinates of **x,** the corresponding **x'** in terms of **x** is

$$\mathbf{x}'(0,\ y'=\sqrt{x^2+y^2},\ z'=z) \tag{3-40}$$

Next, the magnetic density **B'**(0, B_y', B_z') at location **x'** is calculated using Equation (3-3) with **H'** obtained from the steps shown in Example 3.2. Finally, **B**' is transformed to **B** at **x**. The result is given by Equation (3-41):

$$\mathbf{B} = \left[B'_x x / \sqrt{x^2 + y^2} \quad B'_y y / \sqrt{x^2 + y^2} \quad B'_z \right]^{\mathrm{T}} \tag{3-41}$$

Once **B** is known, the force on a body can be computed from the surface integration of the Maxwell stress tensor using Equation (3-17).

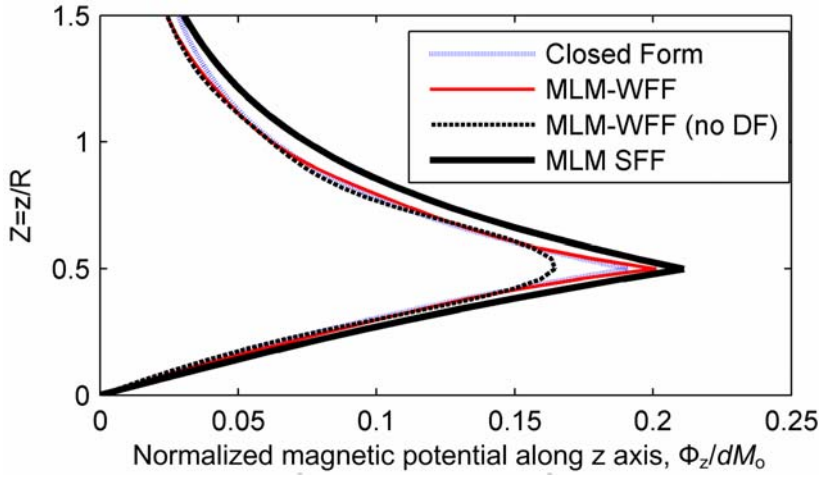
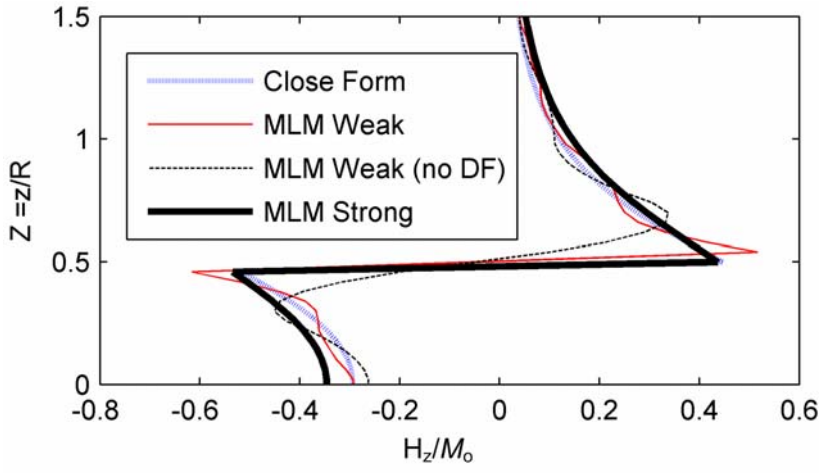
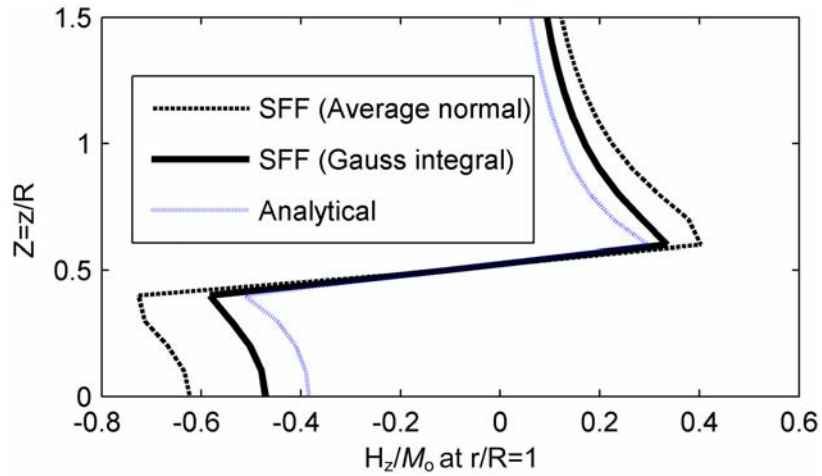


Figure 3-5 Comparison of magnetic potential along the z axis



(a) Along the z-axis (*r/R*=0)



(b) Along the z-direction at *r/R*=1

Figure 3-6 Results of computed field intensity

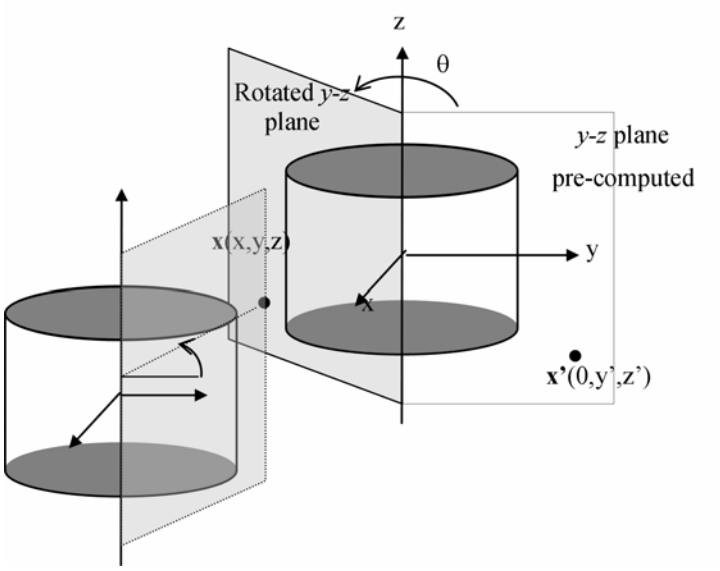


Figure 3-7 Schematic illustrating the overall field computation

Example 3.4

We verify experimentally the magnetic force computation using the setup shown in Figure 3-8. The pair of permanent magnets is separately mounted on two cantilever beams, one of which is driven by a precision NSK ball-screw while the other carries a strain-gage that measures the repulsion force. The computed forces are compared in Figure 3-9 against the experimental data where the values characterizing the two identical magnets are listed in Table 3-3. The effect of the corner on the magnetic force computation using MLM-SFF is shown in Figure 3-10, where the errors (with respect to the analytical solution) in the force computed using two different methods are compared. The computation using the Gauss integral reduces the uncertainty at the corner where the normal is not well defined.

Table 3-3: Parameters used in Example 3.3

Parameters	Value
Magnetic Radius R (mm)	6.35
Magnetic thickness d (mm)	6.35
Magnetization $\mu_0 M_0$ (Tesla)	1.35
Air gap g (mm)	0.5

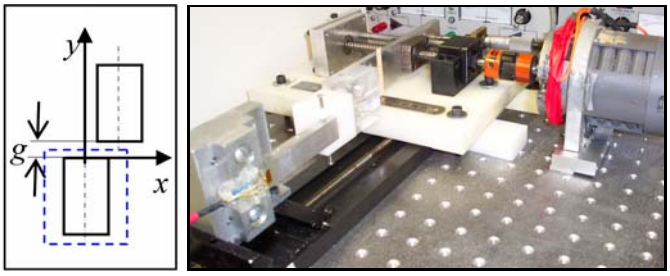


Figure 3-8 Experimental setups [124]

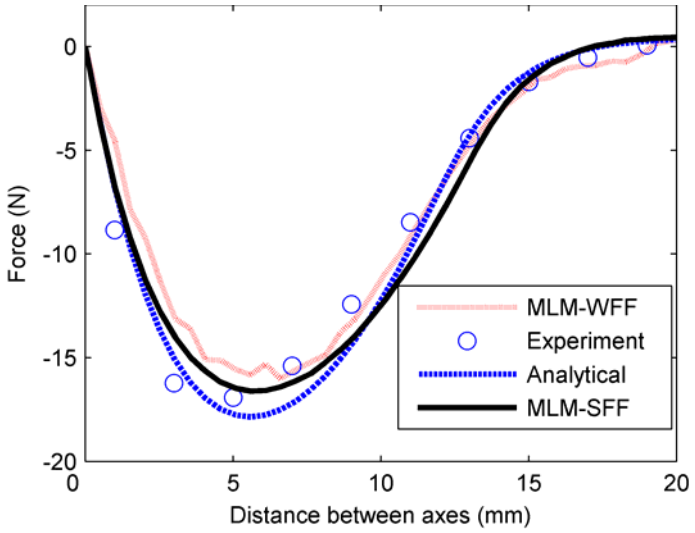


Figure 3-9 Repulsion force as a function of displacement

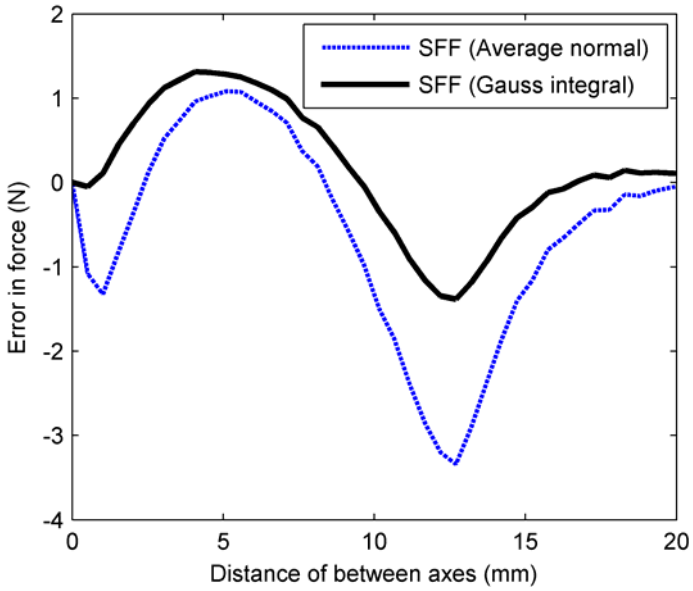


Figure 3-10 Effect of corner on MLM-SFF

3.5 Summary

The effects of two different discretization methods, strong and weak form formulation, on the computational accuracy and efficiency of computing electromagnetic fields have been investigated in this chapter. The computational results were also verified by comparing the computed magnetic force between two permanent magnets against those measured experimentally. Both methods converge to exact solutions with increasing numbers of nodes. However, the computation time of MLM is significantly higher than FEM due to the highly nonlinear shape function. While MLM-WFF offers better overall accuracy, it requires special treatment to approximate the discontinuity of the field intensity, which exhibits oscillations near the discontinuous interface. The discontinuity function added to the MLM-WFF improves the accuracy of the prediction but can not remove the oscillations completely. We also show the problem of un-defined normal at the corner in MLM-SFF and demonstrate the use of the Gauss Integral law to

alleviate the problem. As MLM-SFF requires no meshes and no integration to derive the numerical model, and the domain discretization and the procedure for imposing the boundary conditions are simple, it represents an interesting trade-off between the Gelerkin-based MLM and FEM.

In the subsequent chapters, we focus on MLM-WFF, since our study shows that MLM-SFF suffers from the low accuracy when irregular node distribution is used.

CHAPTER IV

ADAPTIVE MESHLESS METHOD

4.1 Overview

In Chapter III, we presented methods for solving magnetic fields with a discontinuous material boundary condition using MLM with pre-assigned nodes. Although, these methods have effectively improved computational accuracy, they need additional effort to setup the material boundary conditions for a given geometry. In this chapter, we introduce an adaptive MLM as an alternative method for solving such problems. Without utilizing special treatment around the boundary region, the adaptive method can automatically locate high gradient regions and insert additional nodes accordingly. Specifically, this chapter presents the following:

1. *An error estimation technique is offered.* We extend to MLM the posteriori error estimation technique originally developed for FEM. This technique was based on the observation that the computational results are more accurate at the nodes than at other locations. The technique has achieved some successes in adaptive FEM [125, 126]. However, the basis function in MLM is, in general, not a polynomial; the posteriori error estimation technique developed for FEM cannot be directly applied to MLM. We present here a modified error estimation built on two different support sizes of a basis function. As will be illustrated with a one-dimensional (1D) example, this modified error estimation characterizes the true error remarkably well, and its computation in MLM is simpler than in FEM.

2. *Practical issues related to the nodal insertion are discussed.* We present an automatic node insertion method based on a Voronoi plot technique along with the partition unity integration [127] scheme for obtaining the discretized system of weak-form formulated equations. While the MLM does not need elements to perform numerical integration as in FEM, most of MLM divide their computational domains into small numerical integration cells (called background cells). When the nodal density increases at a local area, the density of these background cells in that area must increase accordingly in order to ensure computational accuracy; this makes MLM lose some of its advantages. To overcome this drawback, we introduce a different numerical integration scheme (called the partition unity integration). In this proposed method, the density of integration cells simply increases with the number of nodes.

3. *The adaptive ML computation is validated.* Three numerical examples (that have exact solutions) are given to validate the computation of adaptive MLM. They also illustrate the processes of error estimation and automatic node insertion, and demonstrate the effectiveness of the adaptive MLM on the convergence. In this paper, the weak form equations are derived using the Galerkin method in the MLM.

4. *Applications for design analysis of electromagnetic actuators are illustrated.* We illustrate the use of adaptive MLM for design of electromagnetic actuators with high coercive permanent magnets. Magnetic forces are computed using the Lorenz force law for two examples. The computational results are compared against a set of published experimental results. Additionally, we show how MLM can be used to improve the torque-to-weight ratio in the pole design of a three DOF spherical motor.

4.2 Error Estimation

One of the most common methods to improve the accuracy of the numerical approximation is to reduce the nodal space (or increase the density of the nodes). The simplest way to do this is to uniformly increase the nodal density in the whole computational domain. However, if large numerical errors occur only in certain local regions, this method is inefficient, since extra nodes in small error regions do not help improve the overall computational accuracy; they would simply lengthen the computation time. Thus, it is desired to have an estimate of the overall error distribution of the computation so that additional nodes can be effectively inserted accordingly into the large error regions.

The exact numerical error e can be defined as follows:

$$e(\mathbf{x}) = \Phi_e(\mathbf{x}) - \Phi_a(\mathbf{x}) \tag{4-1}$$

where Φ_e and Φ_a are the exact potential field distribution and the approximated solution of the MLM respectively. However, Φ_e is often unavailable in practice. Thus, a modified form is used to estimate the numerical error:

$$\hat{e}(\mathbf{x}) = \Phi_h(\mathbf{x}) - \Phi_l(\mathbf{x}) \tag{4-2}$$

where Φ_h and Φ_l are both numerical results but Φ_h is more accurate than Φ_l. As an example, Φ_l is a solution obtained with $n \times n$ number of nodes and Φ_h with $2n \times 2n$ nodes. However, it is desired that Φ_h be computed without a recalculation that uses a denser number of nodes for computational efficiency.

We present an alternative error estimation based on two different support sizes of a basis function to locate regions of large numerical errors for the adaptive MLM:

$$\tilde{e}(\mathbf{x}) = \sum_{i=1}^{n} \Psi_{i,d}(\mathbf{x})\Phi_{i,d} - \sum_{i=1}^{n} \Psi_{i,2d}(\mathbf{x})\Phi_{i,2d} \tag{4-3}$$

where $\Psi_{i,d}$ and $\Psi_{i,2d}$ denote the basis functions at the i^{th} node with a support size d and $2d$ respectively; $\Phi_{i,d}$ is the solution solved in the previous computation step; and $\Phi_{i,2d}$ is the fitted result using the basis function with a support size of $2d$**.** As an example, we consider the reproducing kernel basis function given in Chapter II. If the basis function is non-interpolating (as often the case in MLM), $\Phi_{i,2d}$ can be solved from the following system of linear equations:

$$\Phi_{2d} = \mathbf{E}^{-1}\Phi_{d} \tag{4-4}$$

where the elements of the matrix **E** are given as

$$\varepsilon_{i,j} = \Psi_{j,2d}(\mathbf{x}_i) \tag{4-5}$$

The rationale for Equation (4-3) can be explained with the aid of Figure 4-1, which compares two different support sizes of an RKP basis function. As shown in Figure 4-1, the larger the support size, the smoother is the basis function. In general, it is more difficult for the basis function with a larger support size to approximate a function with an abrupt change in the solution. Thus, regions of large errors can be characterized by comparing the approximation solutions obtained using the two different basis functions. Numerical experiments have confirmed this finding.

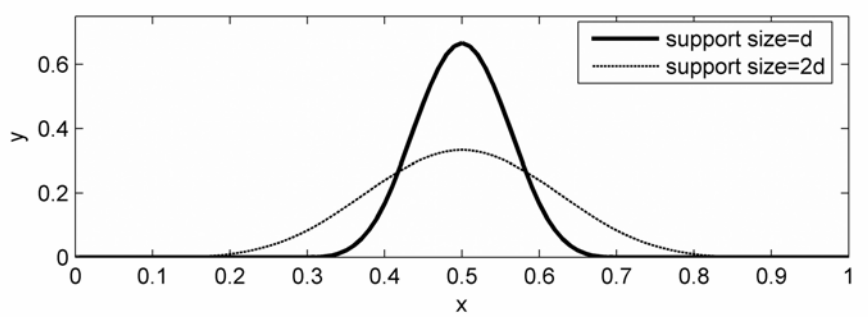


Figure 4-1 RKP basis function with two different support sizes

Once the error is estimated from Equation (4-3), locations of large errors are identified as follows:

$$\forall \mathbf{x}_a : \tilde{e}(\mathbf{x}_a) > e_p \tag{4-6}$$

where x_a is the test location; and e_p is a specified error threshold. We illustrate the error estimation with an example.

Example 4.1: 1D problem illustrating the error estimation for MLM

We illustrate the method here using a 1D problem characterized by the 2nd order ordinary differential equation:

$$d^2y/dx^2 = f(x) \tag{4-7}$$

where $f(x) = -6x - e^{-(2x-1)^2/4\alpha^2}\left[2-(2x-1)^2/\alpha^2\right]/\alpha^2$. The boundary conditions are

$$y(0) = e^{-1/4\alpha^2} \text{ and } y(1) = -1 + e^{-1/4\alpha^2}$$

where α is a constant used to control the shape of the solution. The exact solution is given by $y = -x^3 + e^{-(2x-1)^2/4\alpha^2}$.

To illustrate the error estimation, we solve Equation (4-7) numerically using MLM with weak form formulation (WFF). The weak form equation is obtained by substituting the ML approximation

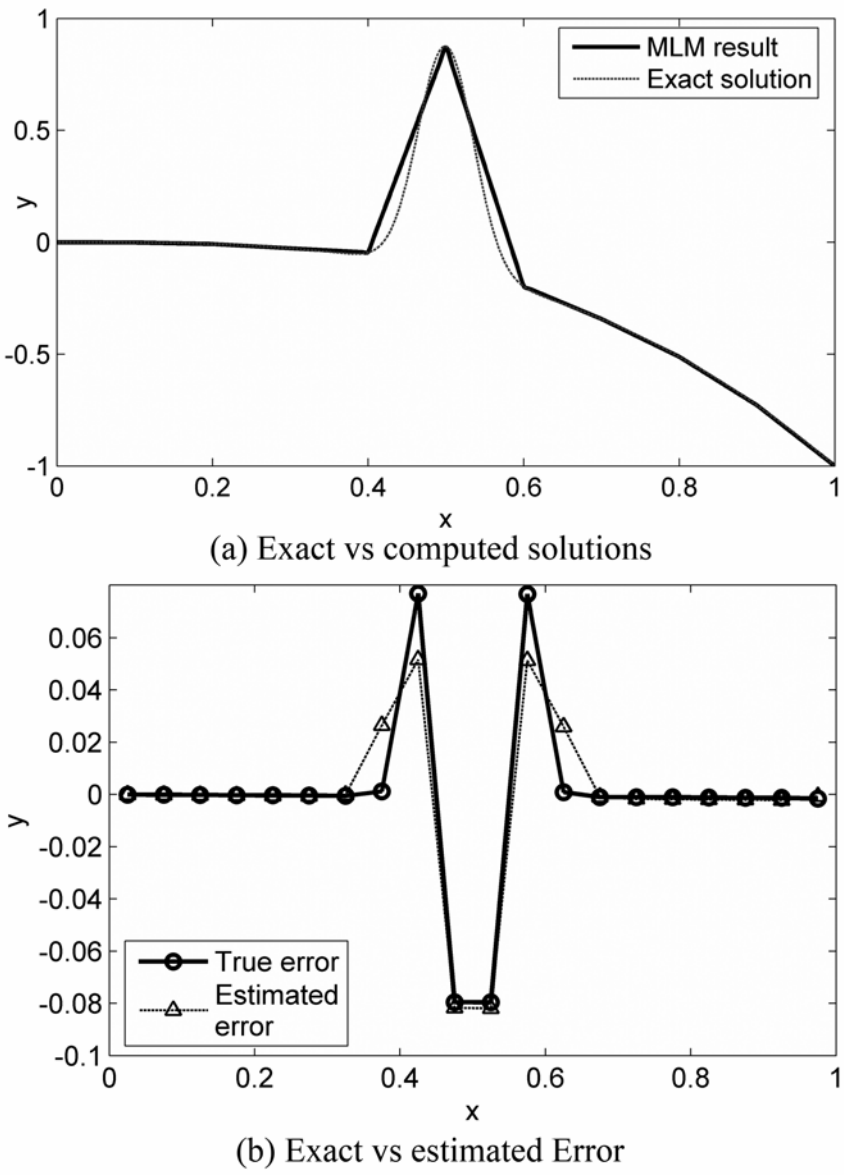
$$\hat{y}(x) = \sum_{i=1}^{n} \Psi_i(x) y_i$$

into Equation (4-7) and integrating the result by parts, which yields

$$\int_0^1 \frac{d\Psi_j}{dx} \sum_{i=1}^{n} \frac{d\Psi_i}{dx} y_i dx = -\int_0^1 \Psi_j f(x) dx \tag{4-8}$$

For the purpose of inserting additional nodes, we compare the estimated error at the mid point between two adjacent nodes against the exact error. As shown in Figure 4-2(a), the solution has a high gradient region around $x=0.5$. Figure 4-2 (b) shows that the results of the MLM (with a uniform distribution of 21 nodes) have a relatively large error around the high gradient region of $x=0.5$.

In order to insert additional nodes efficiently, the error estimation must identify this large error region faithfully with reasonable accuracy. As compared in Figure 4-2 (b), the estimated error characterizes the true error remarkably well, and its computation in MLM is simpler than in FEM.



(a) Exact vs computed solutions

(b) Exact vs estimated Error

Figure 4-2 Comparison between exact and estimated errors

In order to minimize computation, the moment matrix **P** is updated using its value from the previous computational step:

$$\mathbf{P}(\mathbf{x}) = \mathbf{P}_o(\mathbf{x}) + \sum_{i=1}^{n_a} \mathbf{h}(\mathbf{x}-\mathbf{x}_i)\mathbf{h}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_i)\Lambda(\mathbf{x}-\mathbf{x}_i) \tag{4-9}$$

where n_a is the number of newly added nodes.

4.3 Node Insertion Scheme

Additional nodes can be inserted into the computational domain using the Voronoi plot technique that constructs one Voronoi cell for each node.

Node Insertion Scheme:

An example Voronoi plot for a 2D computational domain is shown in Figure 4-3, where the solid dots represent the nodes and the dashed lines are boundaries of the Voronoi cells. As shown in Figure 4-3, a Voronoi cell is a polygon containing all the points closest to the node that it surrounds. The error at the vertices of each Voronoi cell is computed from Equation (4-3). If the error at a corner point satisfies Equation (4-6), a new node is created at that point as illustrated in Figure 4-3. The three triangles at the corners of a Voronoi cell are example regions of large numerical errors.

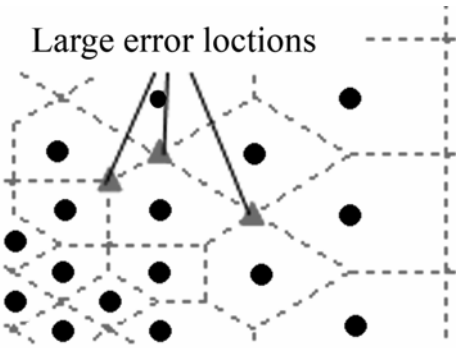


Figure 4-3 Voronoi plot with 3 large error points

Support Size:

The support size of the inserted node is calculated using Equation (4-10) as the maximum distance from the node to the surrounding node whose Voronoi cell is adjacent to this node:

$$r_i = a_p \bullet max\left(\left\|\mathbf{x}_j - \mathbf{x}_i\right\|\right) \tag{4-10}$$

where r_i is the support radius for i^{th} node; $\mathbf{x}_i$ and $\mathbf{x}_j$ are the coordinates of i^{th} and j^{th} nodes respectively. The Voronoi cell of the j^{th} node is adjacent to the Voronoi cell of the i^{th} node. In Equation (4-10), a_p is a constant coefficient normally taken a value between 1 to 3. For the newly inserted node, the support size of its basis function must be chosen carefully considering the following trade-offs:

1) The support radius must be sufficiently large to cover enough nodes for constructing the ML basis function. On the other hand, it is desired to localize the effect of the newly inserted nodes, and thus the support radius should be kept small.

2) Computational load increases as the support radius increases.

Partition unity integration:

The partition unity integration performs the numerical integration based on the support size of the basis function. When a new node is inserted, a new integration cell is automatically created as illustrated in Figure 4-4 and thus, this numerical integration scheme is very suitable for adaptive computation.

Most of the basis functions used in MLM (including RKP basis function) have the partition unity property which is expressed as Equation (2-22).

With this property, the integration for an arbitrary function $f(\mathbf{x})$ in the computational domain can be computed as follows:

$$\int_\Omega f(\mathbf{x})d\mathbf{x} = \int_\Omega f(\mathbf{x})\sum_{i=1}^{n}\Psi_i(\mathbf{x})dx = \sum_{i=1}^{n}\int_\Omega f(\mathbf{x})\Psi_i(\mathbf{x})dx \tag{4-11}$$

where Ω is the computational domain. To exclude points outside the computational domain, Equation (4-11) is written such that the integration is within the support domain S_i of i^{th} basis function:

$$\sum_{i=1}^{n}\int_{\Omega} f(\mathbf{x})\Psi_i(\mathbf{x})dx = \sum_{i=1}^{n}\int_{S_i} f(\mathbf{x})P(\mathbf{x})\Psi_i(\mathbf{x})dx \tag{4-12}$$

where $$P(\mathbf{x}) = \begin{cases} 1 & when\ \mathbf{x} \in \Omega \\ 0 & when\ \mathbf{x} \notin \Omega \end{cases}$$

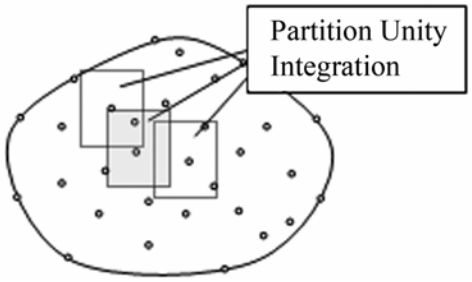


Figure 4-4 Partition Unity Integration cells

The global integration for the whole computational domain is divided into *n* sub-integration domains and performed upon the support domain of *n* basis functions. Because the support domain of the basis functions, in general, has a regular shape, a conventional numerical integration scheme such as Gaussian quadrature can be applied easily.

Example 4.2: Effect of adaptive node insertion on high gradient

Figure 4-5 visually demonstrates the converging process of the adaptive MLM for the 1D example problem given in Example 1 (Figure 4-2), where the initial computation (of 11 uniformly distributed nodes) was compared with the exact solution. At the end of each computation, the error at the mid point of every two adjacent nodes is estimated,

which is shown in Figure 4-5 (a). New nodes are inserted at regions where the estimated errors exceed a predefined threshold. In this research, the average of the error distribution is used as the error threshold. The results of automatic node insertion are given in Figure 5(b). As shown in Figure 4-5, all the inserted nodes concentrate around the high gradient region indicating that the automatic node insertion correctly locates the large error region. We then apply the method of partition unity integration to Equation (4-8):

$$\sum_{k=1}^{n} \oint_{L_k} \Psi_k P(x) \frac{d\Psi_j}{dx} \sum_{i=1}^{n} \frac{d\Psi_i}{dx} y_i dx = \sum_{k=1}^{n} \oint_{L_k} \Psi_k P(x) \Psi_j f(x) dx \tag{4-13}$$

As seen in Figure 4-5(c) and (d), where the support radius of 1.2 is chosen for the adaptive computation, the results of the adaptive MLM converge to the exact solution rapidly after two adaptive computations, each of which uses 4 additional nodes.

Example 4.3: Effect of adaptive node insertion on converging speed

Consider the 2D problem $$\nabla^2 u = \nabla^2 f(x,y) \tag{4-14}$$

with the following boundary conditions:

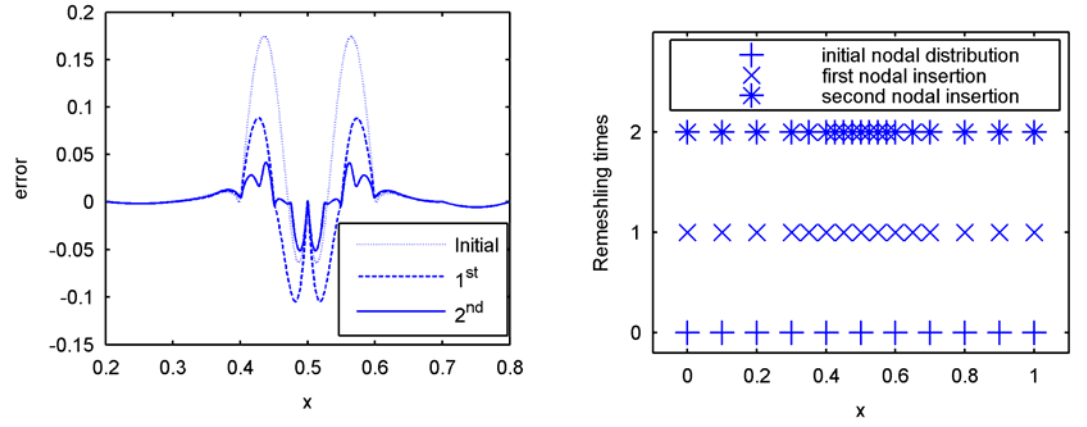
$$u(x,0) = 0\,;\; u(x,1) = 0\,; u(0,y) = 0\,;\; u(1,y) = 0 \tag{4-15}$$

The exact solution is given by

$$f(x,y) = 5x^2y^2(1-x)^2(1-y)^2(e^{10x^2}-1)(e^{10y^2}-1) \tag{4-16}$$

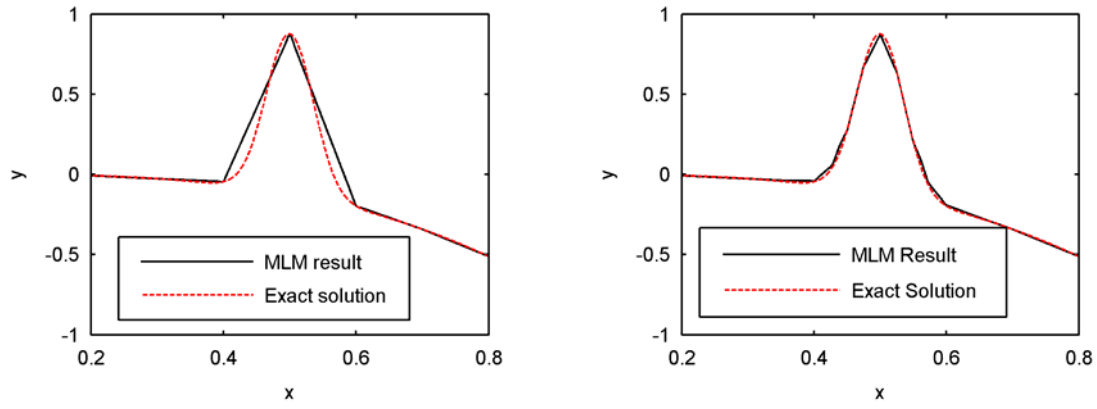
We investigate here the effects of the adaptive node insertion on the convergence speed by comparing it to a commonly used weak-form-formulated MLM (with a globally uniform distribution of nodes). In other words, the nodes of the uniform-node MLM are increased uniformly in the computational domain while the adaptive MLM increases its

nodes according to the estimated errors. Both methods start with an initial computation of 6×6 nodes. Three successive insertions are performed for each method. For the uniform-node MLM, the three successive node distributions are 8×8, 9×9 and 13×13. The node insertion of the adaptive MLM is automatically generated using the estimated error criterion Equation (4-6) and is demonstrated in Figure 4-6.



a) Computational error

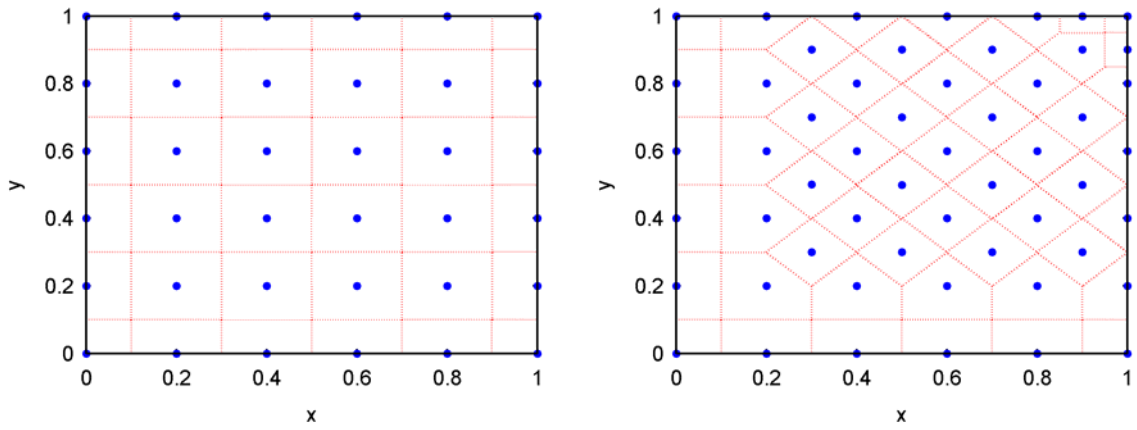
b) Nodal insertion process



c) 1st adaptive (15 nodes)

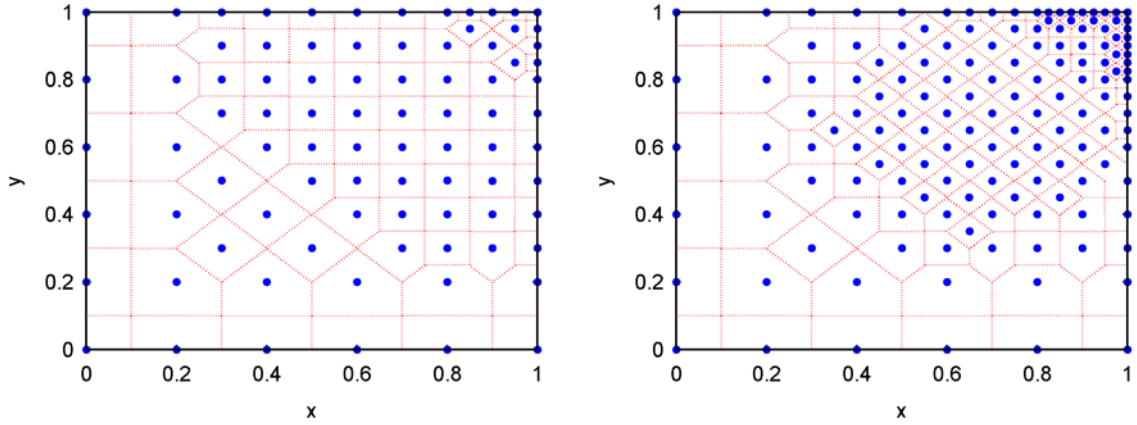
d) 2nd adaptive (19 nodes)

Figure 4-5 Converging process of a 1D problem



a) initial nodal distribution (36)

b) 1st nodal insertion (54)

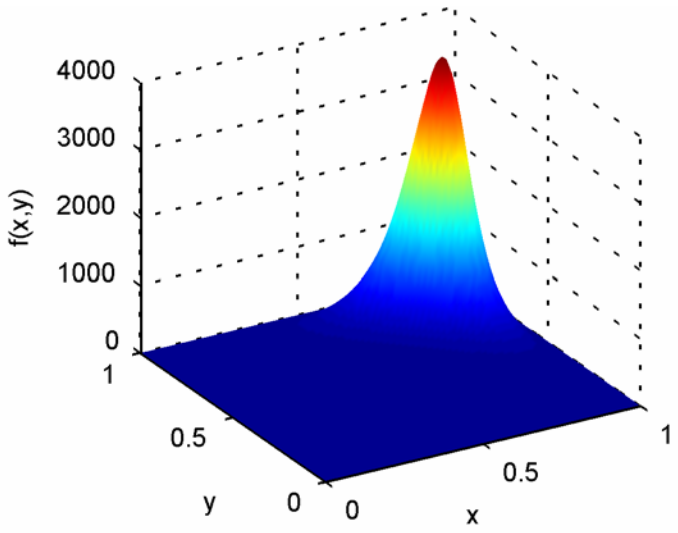


c) 2nd nodal insertion (84)

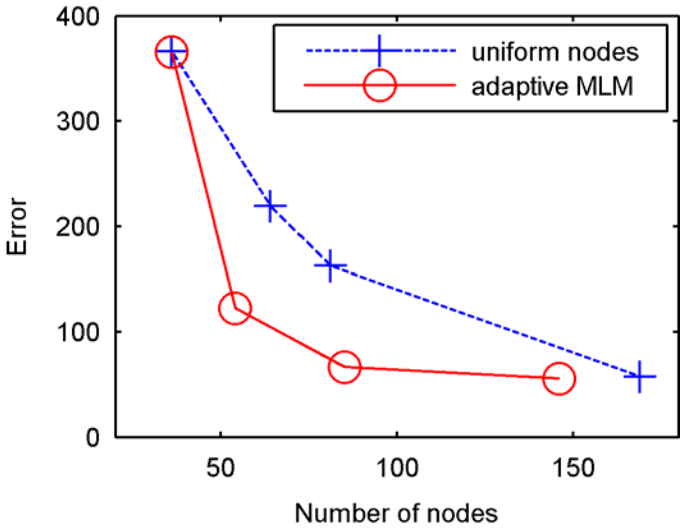
d) 3rd nodal insertion (146)

Figure 4-6 Process of adaptive nodal insertion

The computation errors defined in Equation (2-39) are plotted in Figure 4-7(b):



a) exact solution



b) error comparison

Figure 4-7 the exact solution of example and error comparison

Since the computational time is directly proportional to the number of nodes, the comparison is made by plotting the computational errors versus the number of nodes in Figure 4-7(b). The following observation can be made concerning the convergence processes of these two methods:

1. Both methods tend to converge to the exact solution as the number of nodes increases. However, the adaptive MLM has a significantly higher convergence rate.
2. The adaptive MLM effectively identifies the large error regions that occur around the regions of high gradient as expected.
3. The first two nodal insertions result in rapid error reduction as compared to the third insertion. As the number of nodal insertions increases, the error caused by highly irregular nodal distribution may gradually outweigh the benefit generated by additional nodes hinting that the number of nodal insertion iterations should not be too high in order to maintain the efficiency of algorithm.

4.4 Applications to Electromechanical (EM) Actuators

The prediction of magnetic forces involved in the design of an electromagnetic (EM) actuator relies on the solution of its magnetic field. Due to the field discontinuity at the material interface, and the fact that the air gap between the stator and rotor poles is often very small as compared to the dimension of the overall field distribution, it is often difficult for the regular MLM with a smooth basis function to achieve satisfactory accuracy around the air gaps where energy conversion takes place. The following example shows how the adaptive MLM can be used effectively to improve the field accuracy around these regions. Specifically, the objectives of this example are as follows:

1. It illustrates the use of the adaptive node insertion to approximate discontinuities around the material interface.

2. For validation of the adaptive MLM computation, the magnetic force between a permanent magnet (PM) and an electromagnet (EM) calculated using Lorenz Law with the computed field is compared against published experimental results.

3. This demonstrates how the adaptive MLM can be used as an effective tool to analyze designs. As an example, we consider the pole design of a spherical motor which is capable of providing three degrees of freedom (DOF) motion in a single joint. By comparing two different designs, we illustrate the effects of design geometry on the torque-to-weight characteristics.

For the above objectives, we use the adaptive MLM to solve for the magnetic field intensity $\mathbf{H} = -\nabla\Phi$ around a cylindrical permanent magnet (uniformly magnetized along its axis, $\mathbf{M} = M_o \mathbf{i}_z$) in free space, where Φ is the magnetic scalar potential. The results provide a basis for computing the magnetic force under the influence of an electromagnet using Lorenz's law given in Equation (3-19). The weak-form formulation for computing magnetic field has been given in Section 3.4.

Examples 4.4: Approximation of a discontinuity magnetic field

The magnetic scalar potential and field intensity along the *z* axis are obtained by the adaptive MLM and compared against the closed form solution obtained from Equations (3-33) and (3-34) in Figure 4-8(a) and Figure 4-8 (b) respectively. In this example, 41×41 nodes are used in the regular MLM. The adaptive MLM starts with a uniform nodal distribution of 221 (uniform 13x17) nodes and increases to 658 nodes after three successive insertions. As in Example 4.2, all the additional nodes are inserted into large gradient regions around the magnet pole.

While the result of the uniform-node MLM offers a very good overall prediction; it exhibits some oscillations near the discontinuous interface. As compared in Figure 4-8, the potential and field intensity computed by the adaptive MLM with only 658 nodes (less than half of the number of nodes used by the uniform–node MLM) matches the closed form solution very well at the material interface. With a higher nodal concentration at the material interface, the adaptive MLM is able to approximate the discontinuity satisfactorily with a continuous basis function. This suggests that the adaptive MLM is a good alternative for solving problems with field discontinuity.

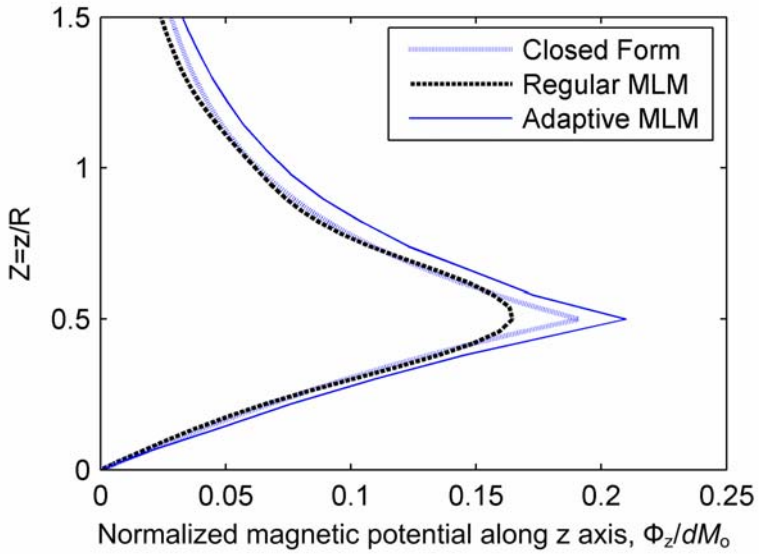
Example 4.5: Force between a permanent magnet and an electromagnet

This example is selected from one of the T.E.A.M. problems [128], where the experimental setup is shown in Figure 4-9 and Table 4-1.

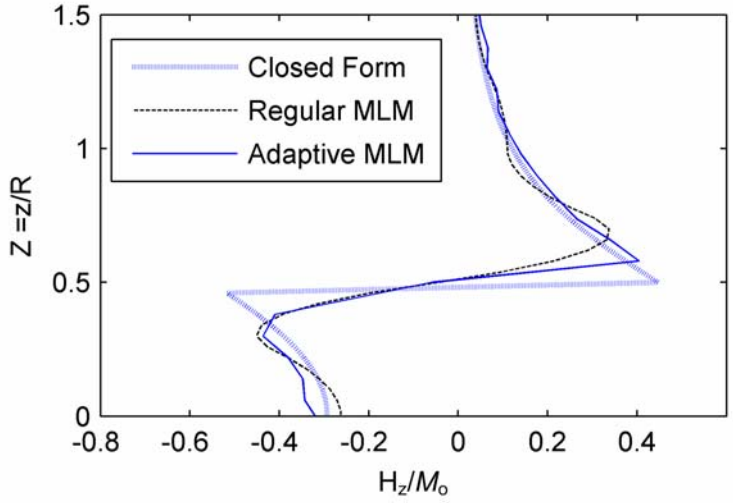
The magnetic force is computed as follows:

1. First, the axis-symmetric magnetic field problem governed by Equations (3-27), (3-28) and (3-29) is solved using adaptive MLM as presented in Sections 4.2 and 4.3.
2. Next, the magnetic force is obtained by conducting integration for the volume of electrical coil using Equation (3-19). The magnetic flux density in Equation (3-19) is computed by following the same computational steps given in Chapter III for Example 3.3.

Figure 4-10 compares the computed forces against published experimental data [128]. The computed restoring force matches the experimental results very well while the computed axial force is slightly larger than the measured force but within 15% difference.



a) magnetic scalar potential



b) magnetic field intensity

Figure 4-8 Effect of adaptive node on handling of discontinuity

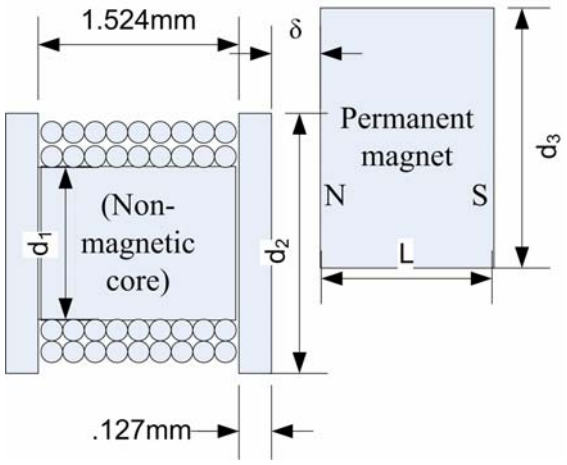
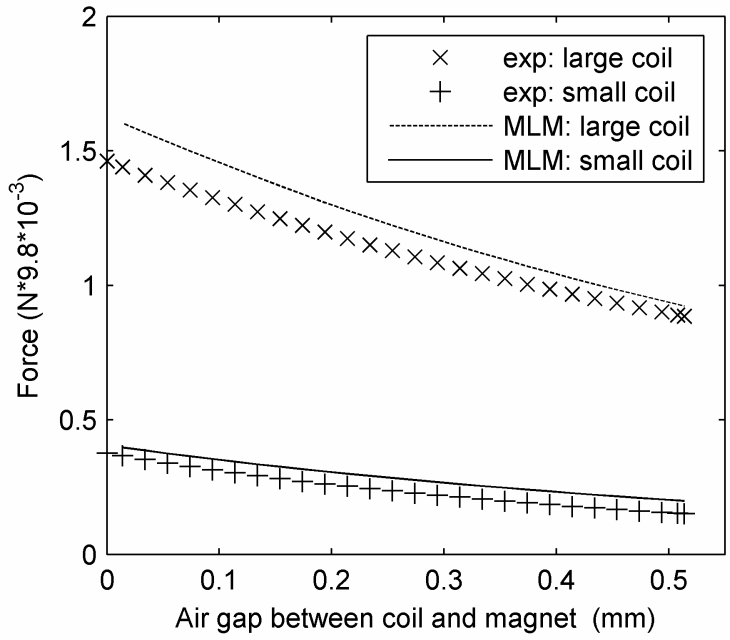


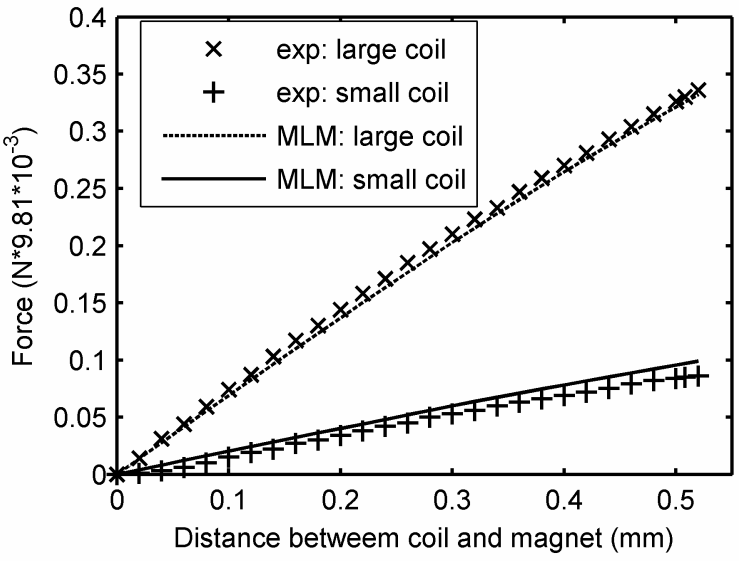
Figure 4-9 Experiment configurations

Table 4-1: Dimensions of Example 4.4

Configurations	Large	Small
d_1 (mm)	3.048	1.524
d_2 (mm)	3.962	3.175
d_3 (mm)	2.998	1.6
L (mm)	1.6	0.8128
Coil res. (Ω)	57	32
Wire length (m)	3	1.68
Samarium-Cobalt magnet; $\mu_0 M_0$=1.02 T		



a) axial force

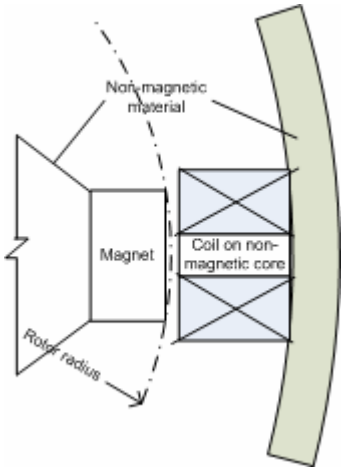


b) restoring force

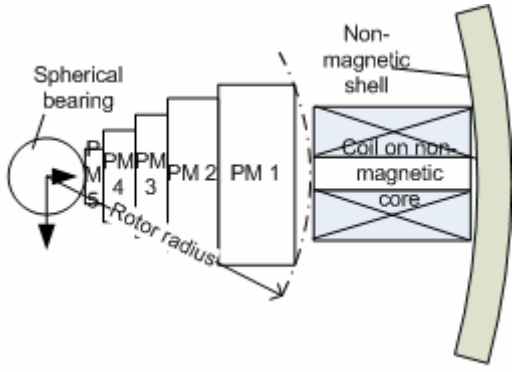
Figure 4-10 Comparison between computed and experimental results

Example 4.6: Pole design of a Three-DOF spherical actuator

Figure 4-11 compares the pole design of two different prototype spherical motors [129, 130]. The geometry and layout of the pole have a significant influence on the torque performance of a spherical motor [131]. Rotor of Design #1 [129] consists of two rows of 8 small PM's whereas Design #2 [130] uses one row of 8 large PM's as rotor poles. Detailed geometries of the two pole designs are given in Table 4-2.



a) design 1 [129]



b) design 2 [130]

Figure 4-11 Comparison of pole designs (not to scale)

Table 4-2: Parameters used in simulation

Design	Rotor radius, mm	Stator Coil OD×ID×L (mm)	# of turns	PM rotor pole OD×L (mm)
1	37.5	19.05×9.53×25.4	1050	12.7×12.7
2	46.5	18×4×27	900	25×10, 20×5, 16×6, 12×3, 8×3

Air gap = .5mm; Magnetization $\mu_0 M_0$=1.27T

The torque between the EM and PM is computed using the Lorenz law from the adaptive-MLM computed field for the two configurations. To provide a common basis for comparison, we compare the torque output per unit radius for a single EM-PM pole-pair. As compared in Figure 4-12, the torque per unit radius for Design #1 is significantly higher although Design #2 uses a much larger PM rotor pole for similar input power. In addition, the compact pole geometry in Design #1 allows for a larger number of rotor poles to be used, which could further improve the torque-to-volume ratio for a specified input power.

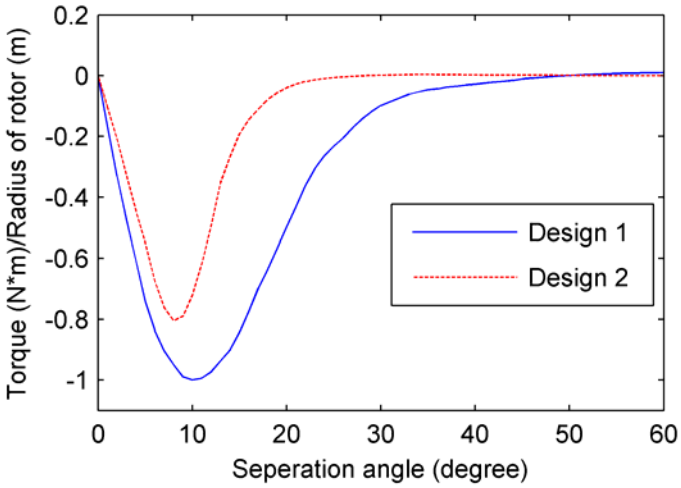


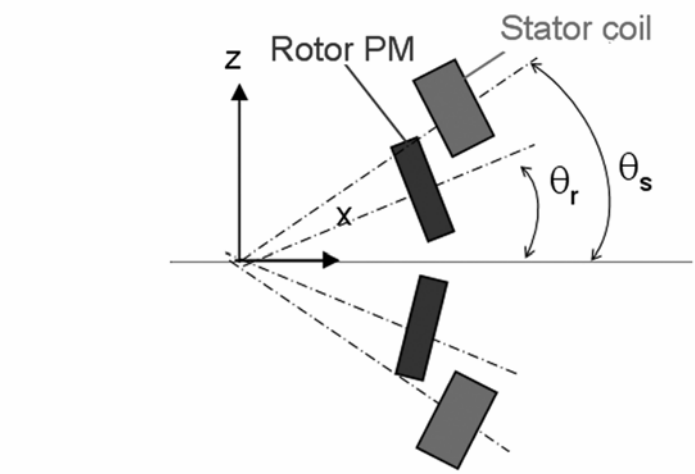
Figure 4-12 Comparison of torque per unit radius

To investigate the effect of the PM-pole layout (denoted as θ_r in Figure 4-13) on the torque profile for Design #1, we compute the torque about the y-axis for three different rotor pole layouts for a specified pair of EM poles spaced at θ_s =26°. The three different layouts are θ_r=0°, 20°, and 25°. For the case θ_r=0°, only one PM that has the same length but twice the volume is used. As shown in Figure 4-13 (b), the pole layout has a significant effect on the torque profile (both its maximum magnitude and the smoothness of its shape). As illustrated in Figure 4-12 and Figure 4-13 (b), the maximum magnitude of the torque and the range of its influence can be doubled but only with a carefully selected set of design parameters. These results illustrate how the adaptive MLM can be effectively used to analyze the effects of pole design on the torque performance of an EM actuator.

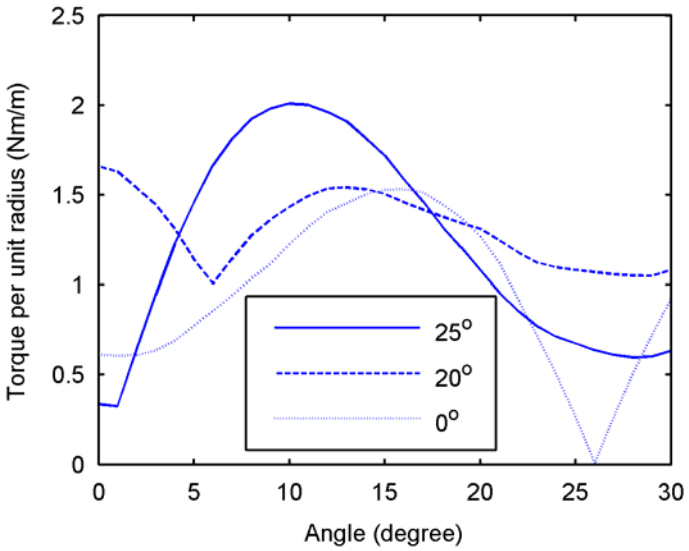
4.5 Summary

In this chapter, a relatively complete adaptive computational method for MLM has been presented and illustrated with several examples. This practical method, which has been validated by comparison with exact solutions, overcomes two technical difficulties associated with MLM; namely, error estimation and nodal insertion. Our results show that the method can faithfully locate large error regions, automatically insert nodes to these regions without human involvement, and improve the computational efficiency significantly. Comparing the converging speed of the adaptive MLM against a regular MLM (with uniform distributed nodes) shows that the adaptive MLM is effective and computationally efficient. Additionally, we compare the electromagnetic forces

computed using the Lorenz law with the field results predicted by the adaptive MLM against published experimental results, which show excellent agreement.



a) pole geometry



b) Torque for different θ_r (θ_s =26°)

Figure 4-13 Effect of pole layout on torque generated

CHAPTER V

NONLINEAR MECHANICAL CONTACT

5.1 Overview

In this chapter, we extend the adaptive meshless method developed in Chapter 4 to solve nonlinear problems of large mechanical deformation contact. In stead of directly using the primary variable as in the linear algorithm, the adaptive procedure in this chapter will utilize the stress information for error estimation. As stated in the first chapter, the adaptive meshless method gains two main benefits when being applied to solve large deformation contact problems. First, the adaptive algorithm will help reduce unnecessary nodes in non-contact regions. Second, when the contact is combined with geometric nonlinearity, the meshless method is able to avoid the computational break-down caused by element distortion as in FEM. Computational examples are provided to illustrate those features.

Specifically, the remainder of this chapter offers the following:

1. We present a general formulation for solving mechanical contact problems involving large deformation, and the numerical method to obtain solutions using MLM. This formulation, which applies the sliding line algorithm [132] along with the penalty method [133] for handling the contact constraints, does not rely on the small (or linear) displacement assumptions commonly made in formulating mechanical contacts. Thus, the solution method presented here is rather general and can be used to solve non-linear contact problems with large deformation.

2. We extend the adaptive MLM to solve non-linear mechanical problems with emphases on contact problems. Unlike [134] where the magnetic problems are formulated in terms of displacements, we derive the error estimation, which identifies regions of large computational errors for automatic node insertion, based on mechanical stresses since large displacement due to rigid body motion does not necessarily result in mechanical stresses.

3. Four examples are given to illustrate the automatic node inserting procedure of the adaptive MLM algorithm and its effectiveness in simulating large mechanical deformation and/or contact. As will be shown, unlike FEM where excessively large deformation could cause severe element distortion and consequently break down the simulation, the adaptive MLM algorithm is able to construct basis functions without using mesh structure.

4. The adaptive MLM algorithm for solving mechanical contact problems has been validated by comparing the MLM computed results against the analytical solution whenever possible, and those simulated using ANSYS (a commercial finite element analysis package).

5.2 Formulation of Mechanical Large Deformation

For static or quasi-static problems involving large deformation, the three governing equations are given by [135]

$$\sum_{j=1}^{3} \frac{\partial P_{ji}}{\partial X_j} + \rho_0 b_{0i} = 0 \quad (i = 1,2,3) \tag{5-1}$$

where ρ_0 and b_0 are the density and body force of the original un-deformed state; and P_{ji} is the element of the 1st Piola-Kirchhorff (PK) stress tensor **P**. For linear, small displacement problems, the Cauchy stress **σ** is used in place of **P**. To solve Equation (5-1) for the displacement function u as an independent variable, the asymmetric stress tensor **P** is transformed to the symmetric 2nd PK stress tensor **S** by

$$P_{ji} = \sum_{r=1}^{3} S_{ir} \frac{\partial x_r}{\partial X_j} \tag{5-2}$$

where S_{ir} is an element of $\mathbf{S} \in R^{3\times3}$ (that is related to the displacement u through a material constitutive model). In this paper, the general Hooke's law is used:

$$S_{ir} = \sum_{k=1}^{3}\sum_{l=1}^{3} C_{irkl}\left(\varepsilon_{kl} + \overline{\varepsilon}_{kl}\right) \tag{5-3}$$

where C_{irkl} is the element of the material compliant tensor **C** (a material property); and $\varepsilon_{kl} + \overline{\varepsilon}_{kl}$ are the terms in the element of the Green strain tensor given by

$$\varepsilon_{kl} = \frac{1}{2}\left(\frac{\partial u_k}{\partial X_l} + \frac{\partial u_l}{\partial X_k}\right) \text{ and } \overline{\varepsilon}_{kl} = \sum_{m=1}^{3} \frac{\partial u_m}{\partial X_k}\frac{\partial u_m}{\partial X_l} \tag{5-4}$$

For linear small displacement problems, the higher order terms in the Green strain tensor can be ignored, or $\overline{\varepsilon}_{kl} \approx 0$; the Green strain tensor reduces to Cauchy strain ε_{kl}.

To solve Equation (5-1), we need the BC's to find a solution that is physically relevant. The two types of boundary conditions (BC's) for a continuum body, the Dirichlet and the Neumann BC's, are the displacement $\overline{u}_i$ and traction $\overline{t}_i$ (or force per unit area) BC's respectively:

$$u_i = \overline{u}_i \quad (i = 1,2,3) \text{ on } \Gamma_u \tag{5-5}$$

$$\sum_{j=1}^{3} P_{ij} n_j = \overline{t}_i \quad (i = 1,2,3) \text{ on } \Gamma_t \tag{5-6}$$

where **n** is the normal vector of boundary.

5.3 Mechanical Contact and Constraint Formulation

Consider two bodies, Ω_A and Ω_B, bounded by Γ_A and Γ_B, respectively as shown in Figure 5-1, where X is the original un-deformed coordinate of a particle; and $\mathbf{x}_A(X,t)$ and $\mathbf{x}_B(X,t)$ represent the deformed coordinate of an arbitrary particle on the bodies A and on B at time t respectively. Physically, contact can be interpreted as a constraint imposed on continuum mechanics implying that the two bodies can not penetrate into each other:

$$\Omega_A \cap \Omega_B = 0 \tag{5-7}$$

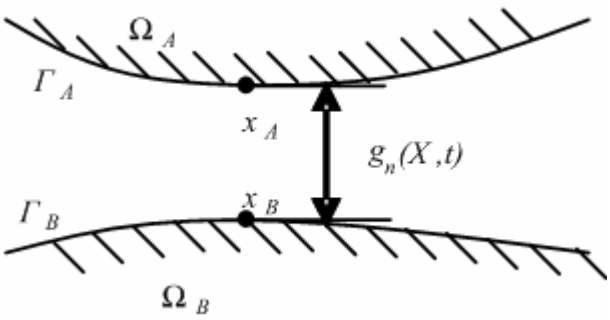


Figure 5-1 Illustration of contact between two bodies

Defining the contact condition

Although the contact constraint Equation (5-7) can be easily understood, it is inconvenient to handle numerically, as computational methods require a discretized form.

Thus, we formulate the contact problem as a displacement constraint posed on discretized nodes. The distance between two particles on Γ_A and Γ_B can be expressed as a gap function $g_n(X,t)$, which obeys the following rules:

$$g_n(X,t)\begin{cases} > 0, & \text{when two points are not in contact,} \\ = 0, & \text{when two points are at contact,} \\ < 0, & \text{penetration occurs.} \end{cases} \tag{5-8}$$

The first two conditions in Equation (5-8) state that the distance between the two points at the same contact should be zero when the two bodies are in contact; or greater than zero when they depart. The last condition in Equation (5-8) is physically invalid since the two bodies can not move into each other. However, small penetration is necessary numerically, as in penalty methods. In formulating the contact problem, the penalty method assumes that the normal component of the contact force τ_{cn} is proportional to g_n:

$$\tau_{cn} = \begin{cases} 0 & g_n \geq 0 \\ k_n g_n & g_n < 0 \end{cases} \tag{5-9}$$

where the penalty proportionality k_n is a very large number. This approximation approaches ideal contact as $k_n \to \infty$. Since g_n is negative when there is a contact, the force vector $\tau_{cn}\mathbf{n}$ points outward at the boundary of the contact object.

Effect of friction

Once the contact force in the normal direction is known, the tangential component of the contact force (or the friction force) t_{ct} can be obtained by the classic Coulomb friction law in Equation (5-10):

$$\begin{cases} \text{Stick occurs if} & 0 < \left|\tau_{ct}\right| \le -\mu\tau_{cn} \\ \text{Slip occurs if} & \left|\tau_{ct}\right| = -\mu\tau_{cn} \end{cases} \tag{5-10}$$

where μ is the friction coefficient. Since there is a need to quantitatively determine the current state of contact (either "stick" or "slip"), we introduce another gap function g_t to depict the distance that the contact point slips for two adjacent time steps. With g_t, τ_{ct} can be computed as follows:

$$\tau_{ct} = \begin{cases} k_t g_t & \mu\tau_{cn} \ge \left|k_t g_t\right| \quad \text{(stick)} \\ \mu\tau_{cn}\,\text{sgn}(g_t) & \mu\tau_{cn} < \left|k_t g_t\right| \quad \text{(slip)} \end{cases} \tag{5-11}$$

where k_t is the tangential penalty parameter. As in the treatment for the normal contact force in Equation (5-9), the first condition in Equation (5-11) does not satisfy Coulomb law exactly since for the "stick" situation $g_t \equiv 0$; as a result τ_{ct} must be zero as well. However, the friction force can be approximately obtained if we allow for a small slipping distance, and as k_t increases Equation (5-11) approaches the ideal Coulomb law. Unlike τ_{cn} which is always negative when there is a contact, the sign of τ_{ct} can be positive or negative depending on the direction of slip.

Discretization of contact for numerical solutions

In the discretized domain, the two contact bodies are referred to here as the slave and the master. The assignment of master and slave are arbitrary and exchangeable. The coordinates of the discrete nodes are defined in Figure 5-2, where $\mathbf{x}_s$ and $\mathbf{x}_c$ are the slave node and the contact point on the master segment respectively; $\mathbf{x}_{m1}$ and $\mathbf{x}_{m2}$ are the two adjacent master nodes; and $\mathbf{x}_{c0}$ is the contact point of the last computational step. In

Figure 5-2, **n** and **t** are respectively the unit normal and tangential vectors at $\mathbf{x}_c$. The vector **t** is can be computed from the master nodes

$$\mathbf{t} = \left(\mathbf{x}_{m2} - \mathbf{x}_{m1}\right)/\ell \qquad \text{where } \ell = \left\|\mathbf{x}_{m2} - \mathbf{x}_{m1}\right\| \tag{5-12}$$

and **n** can then be obtained from the orthogonality $\mathbf{n} = \mathbf{e}_z \times \mathbf{t}$ where $\mathbf{e}_z$ is an unit vector along the *z* axis.

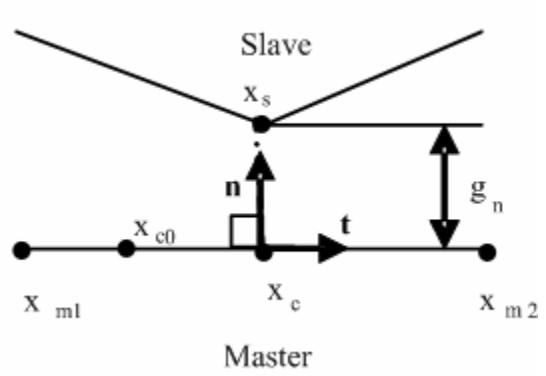


Figure 5-2 Contact gap-function between two discretized bodies

The normal gap function g_n, which is defined as the distance from the slave node to the master segment, can be computed as follows:

$$g_n = (\mathbf{x}_s - \mathbf{x}_{m1}) \bullet \mathbf{n} \tag{5-13}$$

Similarly, the tangential gap function g_t, which is the distance between $\mathbf{x}_c$ and $\boldsymbol{x}_{co}$, can be computed as

$$g_t = (\mathbf{x}_c - \mathbf{x}_{c0}) \bullet \mathbf{t} \tag{5-14}$$

5.4 Weak Form Formulation for Large Deformation Contact

The basic MLM approximation form for an unknown displacement function $u(\mathbf{X})$ can be written as:

$$u(\mathbf{X}) = \mathbf{x} - \mathbf{X} = \sum_{i=1}^{n} \Psi_i(\mathbf{X}) u_i \tag{5-15}$$

where u_i is the nodal control value associated with the i^{th} node; and $\Psi(\mathbf{X})$ is a ML basis function that can be constructed, for example, using the reproducing kernel method [136]. If the ML basis function at the i^{th} node is an interpolating function, u_i is the displacement at this node $u_i=u(\mathbf{X}_i)$. Otherwise, $u_i \neq u(\mathbf{X}_i)$.

The large deformation problem is numerically formulated in weak form. For this, we multiply both sides of Equation (5-1) by the ML basis functions, and integrate the resulting equation by parts, which leads to the following governing equation in weak form:

$$\int_{\Omega_0} \sum_{j=1}^{3} \left(\frac{\partial \Psi_k}{\partial X_j} P_{ji} \right) d\Omega_0 - \int_{\Omega_0} \Psi_k \rho_0 b_i d\Omega_0 - \int_{\Gamma_0} \Psi_k \bar{t}_i d\Gamma_0 = 0 \tag{5-16}$$

Note that Ψ_k is the basis function at k^{th} node.

We formulate the contact problem using the penalty method as follows:

$$\delta W_i = \delta W_e + \delta G_p \tag{5-17}$$

where W_i and W_e are the virtual internal and external works without contact constraint; G_p is the virtual work contributed by the contact force τ_{cn} and τ_{ct} and

$$G_p = \int_{\Gamma_c} \tau_{cn} g_n + \tau_{ct} g_t d\Gamma \tag{5-18}$$

The variations of W_i and W_e are given by:

$$\delta W_i = \int_{\Omega_0} \frac{\partial \Psi_I}{\partial X_j} P_{ji} \delta \mathbf{x}_I d\Omega_0 \tag{5-19}$$

and

$$\delta W_e = \int_{\Omega_0} \Psi_I \rho_0 b_i \delta \mathbf{x}_I d\Omega_0 + \int_{\Gamma_0} \Psi_I \bar{t}_i \delta \mathbf{x}_I d\Gamma_0 \tag{5-20}$$

Incorporating the assumption Equation (5-9) and Equation (5-11) in the penalty method, the variation of G_p becomes

$$\delta G_P = \int_{\Gamma_c} \tau_{cn} \delta g_n + \tau_{ct} \delta g_t d\Gamma \tag{5-21}$$

where the variation of the gap functions can be derived from Equation (5-13) and Equation (5-14):

$$\delta g_n = \left[\mathbf{n}, \quad -(1-\alpha)\mathbf{n}, \quad -\alpha \mathbf{n}\right]^{\mathrm{T}} \bullet \delta \mathbf{x}_w$$

and

$$\delta g_t = \frac{\ell}{\ell_o}\left[\mathbf{t}, \ -\frac{g_n}{\ell}\mathbf{n} - (1-\alpha)\mathbf{t}, \ \frac{g_n}{\ell}\mathbf{n} - \alpha \mathbf{t}\right]^{\mathrm{T}} \bullet \delta \mathbf{x}_w$$

where $\delta \mathbf{x}_w = \left[\delta \mathbf{x}_s, \delta \mathbf{x}_{m1}, \delta \mathbf{x}_{m2}\right]^{\mathrm{T}}$; $\alpha = \left(\mathbf{x}_s - \mathbf{x}_{m1}\right) \bullet \mathbf{t} / \ell$; and ℓ and ℓ_o are the current and previous distances defined in Equation (5-12).

In summary, the governing equations in weak form can be obtained by substituting Equation (5-19), Equation (5-20) and Equation (5-21) into Equation (5-17). As shown in Equation (5-1), Equation (5-2) and Equation (5-3) the 1st PK stress tensor **P** is a nonlinear function of displacement *u*. Thus for the case of large deformation, the discretized governing equations in weak form are a set of nonlinear system equations, the

solution of which can be obtained by applying Newton method. The natural (or Neumann) boundary conditions are applied in the process when the governing equations are converted to weak form. The essential (or Dirichlet) boundary conditions are applied before solving the linearized sets of weak-form governing equations.

5.5 Adaptive Error Estimation for Mechanical Deformation

In Chapter 4, the adaptive error estimation was based on a displacement variable. However, displacement regardless of its magnitude may simply be a result of a rigid body is motion, does not necessarily induce stresses or stains in the mechanical body. Thus, the stress or stain is the more appropriate quantitative variable for error estimation in solving mechanical problems.

The mechanical stress is a 9-component tensor σ_{ij} (or S_{ij} in the case of large deformation) which in matrix form can be represented as a 3x3 symmetric matrix. The three principal components of stress, which are usually used as criteria to determine material failure, are the eigenvalue of the stress matrix. They are coordinate independent, and can be utilized to locate the region of high stresses. The overall magnitude of the stress T_{in} can be written as:

$$T_{\text{in}} = \sum_{i=1}^{3} \lambda_i^{\,2} \tag{5-22}$$

where λ_i is the eigenvalue of stress matrix. In practice, T_{in} is computed from:

$$T_{\text{in}} = \left[\sum_{i=1}^{3} \sigma_{ii}\right]^2 + 2\sum_{i=1}^{3}\sum_{j=1}^{3} \kappa_{ij}\left(\sigma_{ij}^{\,2} - \sigma_{ii}\sigma_{jj}\right) \tag{5-23}$$

$$\text{where } \kappa_{ij} = \begin{cases} 1 \; when \; i \neq j \\ 0 \; when \; i = j \end{cases}$$

The error estimation for inserting additional nodes in solving mechanical problems can be executed as follows:

Step 1. Determine an appropriate support size for the ML basis function.

Step 2. Compute the displacement field $u(\mathbf{X})$ with the original basis function.

Step 3. Fit the displacement result using the basis function but a larger support size.

Step 4. Compute the stress field $\sigma(x)$ from the linear or nonlinear strain Equation (5-4) using the original and the new displacements.

Step 5. Compute T_{in} for the original and the new results.

Step 6. Estimate the error as the difference between two stress magnitudes.

5.6 Numerical Examples

We illustrate here four numerical examples. The first two examples validate the adaptive MLM for a general mechanical deformation problem and a mechanical contact problem respectively. In the 1st example, the adaptive MLM is utilized to compute the deflection of a cantilever beam. Its convergence process is demonstrated by comparing its result with the solution obtained by FEM with a high density mesh. The 2nd example, where the analytical solution is available, is to validate the adaptive MLM along with the contact algorithm. We also compare the results against the solution obtained from commercial FEM code (ANSYS). The 3rd example investigates the effect of friction for a

snap-fit mechanism. The 4th example shows the potential of MLM in medical surgery applications.

Example 5.1: Adaptive MLM for Computation Mechanics

To illustrate the error estimation and node insertion in the adaptive MLM computation, we consider a 2D finger with one of its ends clamped and a vertical shear force P applied at the other free end. Since the exact solution for large deformation is not available for comparison, FEM is chosen here as a basis for illustration. We compare MLM results against those computed using commercial (ANSYS) FEM code, for which condensed 288 2nd order elements (937 nodes) are used to ensure the accuracy of the FEM results. Figure 5-3 shows the parameters used to characterize the un-deformed finger, and a typical deformed shape of the FEM-meshed finger. The material properties (the Young's modules E and the Poisson's ratio μ) and geometry of the beam are given in Table 1.

The initial uniform distribution of 7×4 nodes is used for the MLM computation. After three adaptive computations, the total number of nodes increases to 99. The difference in the y-displacements between the adaptive MLM and the FEM solution was computed for each of the iterations. The % error in Figure 5-4 is defined as $\text{error} = 100\% \times \left(y_{MLM} - y_{FEM} \right) / y_{FEM}$. As shown in Figure 5-4, large % errors are primarily located near x=0 where the beam is clamped. Figure 5-5 shows a snap shot of the node distributions after two adaptive computations, where "•" and "×" denote the original and adaptive inserted nodes respectively. As expected, new nodes are automatically inserted to the regions near the clamped end of the beam. This implies that

the adaptive algorithm correctly identifies the large error regions, then inserts new nodes accordingly to those regions, and effectively reduces the computational errors.

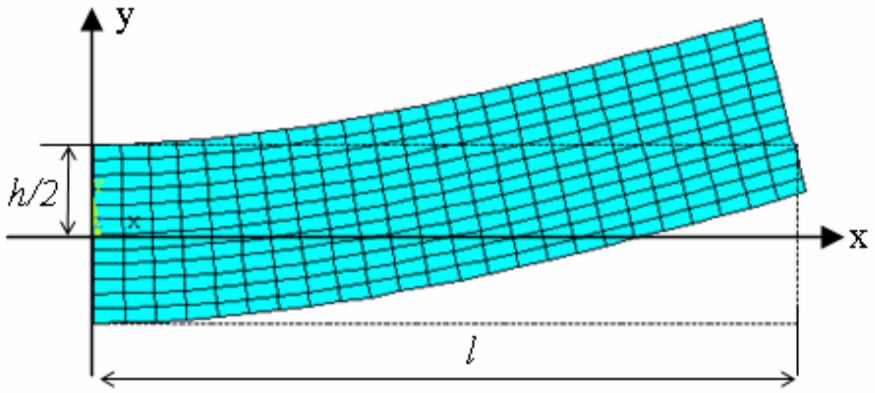


Figure 5-3 FEM mesh and its deformed result (ANSYS)

Table 5-1: Parameters for example 5.1

l (m)	h (m)	E (Pa)	μ	P (N)
48	12	30E6	0.3	1E6

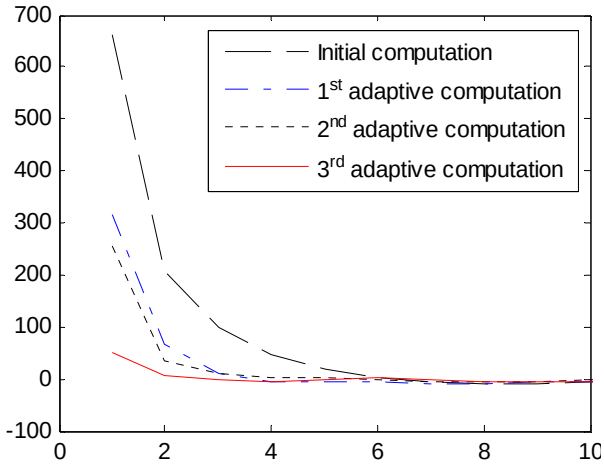


Figure 5-4 Percentage error of MLM for four consecutive adaptive computations

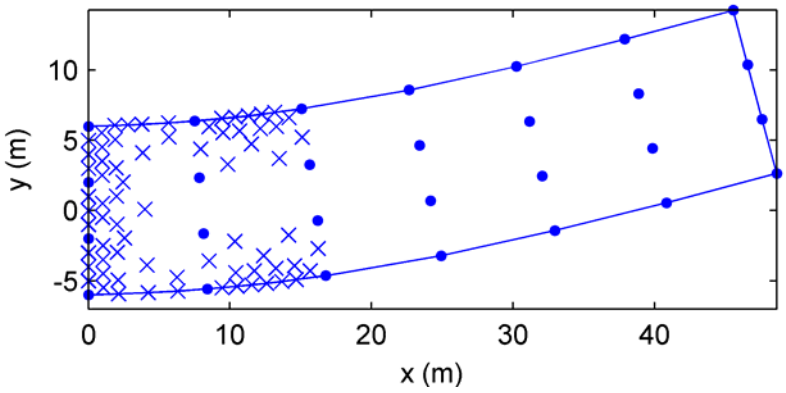


Figure 5-5 MLM nodes after the final adaptive computation

Example 5.2: Contact between rigid and elastic objects

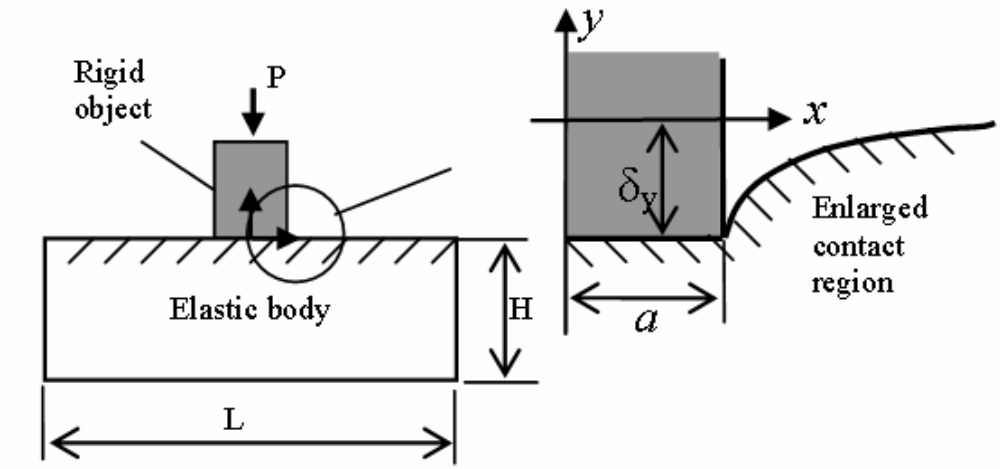


Figure 5-6 Rigid punch contacts with elastic foundation

Table 5-2: Geometry parameters of example 5.2

L (m)	H (m)	a (m)	δ_y
.08	.04	.0025	.0001

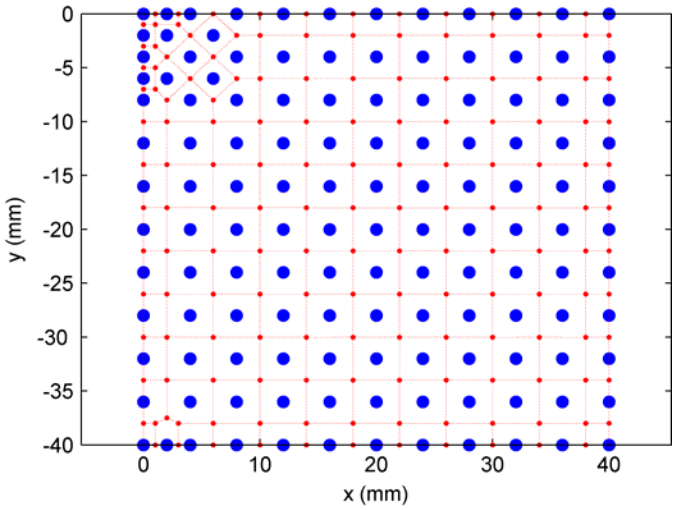
Figure 5-6 shows schematically a classic two-body contact problem, where a small rigid object (which may be a rigid punch or robotic finger) is driven normally into an elastic body. Both objects are infinite in the z-axis. The structure is symmetric with

respect to the *y*-axis, thus only half of the geometry on the positive *x*-axis is solved. The closed form analytical solution describing the displacement along the *y*-direction for the frictionless case can be found in [137].

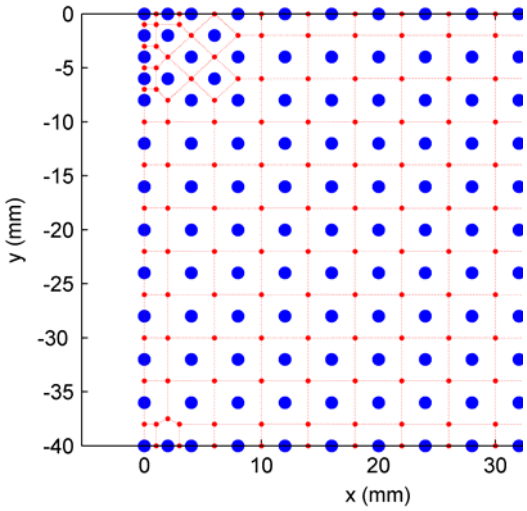
$$u_y(x) = \delta_y - \begin{cases} 0 & \text{when } x \le a \\ \dfrac{2(1-\nu^2)P}{\pi E} \ln\left[\dfrac{x}{a} + \sqrt{\dfrac{x^2}{a^2} - 1}\right] & \text{when } x > a \end{cases} \tag{5-24}$$

where u_y is the displacement in the *y*-direction; P is the force applied on rigid punch; a is the half width of rigid punch; and δ_y is the distance that the rigid object moves into the elastic body.

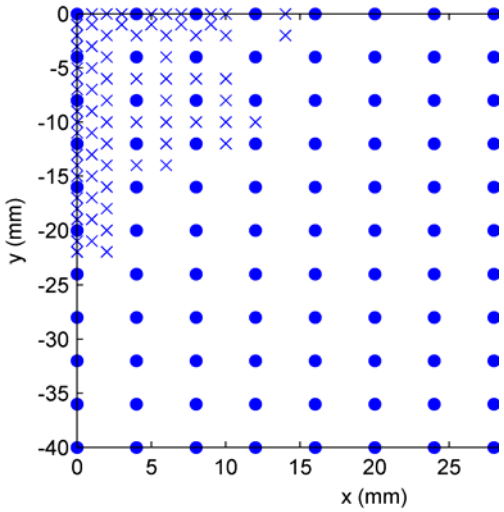
To demonstrate the effectiveness of the adaptive method, no special node refinement is made around the contact region, and the computation starts with a uniform distribution of 11×11 nodes. After three successive computations, the total number of nodes increases from its initial 121 nodes to 194. Figure 5-7(a) and (b) show the Voronoi diagrams of the initial and 2nd node distributions. The final node distribution is shown in Figure 5-7 (c). As illustrated in Figure 5-7, the adaptive algorithm effectively identifies the contact region, and automatically inserts additional nodes around the contact region. The MLM and FEM results are compared against the analytical solutions in Figure 5-8, where FEM uses a total of 544 nodes with special refinement around contact area.



a) Initial Voronoi diagram (• node; …vertexes of a Voronoi cell)



b) 2nd Voronoi diagram



c) Final node distribution

Figure 5-7 Adaptive nodes insertion

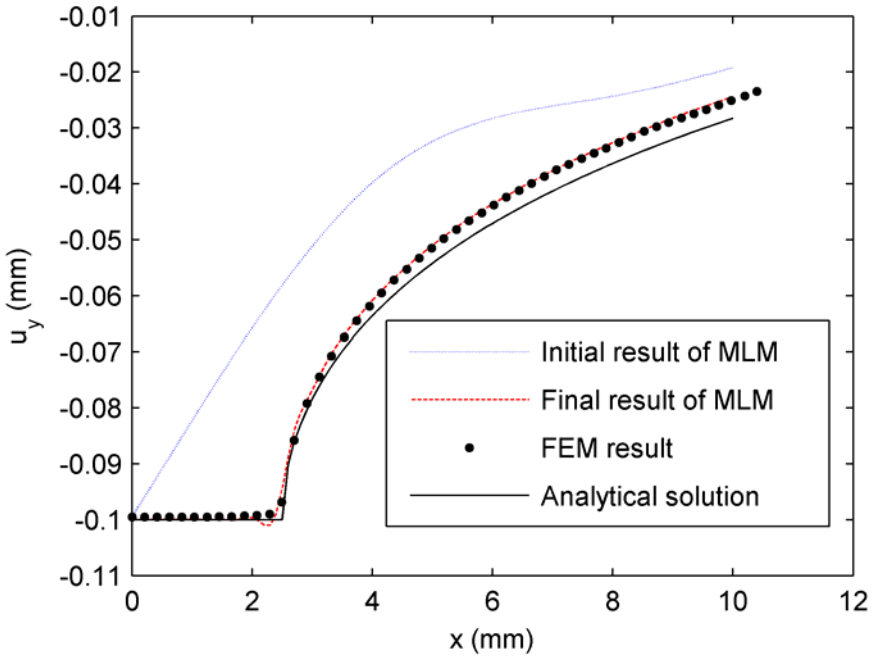


Figure 5-8 Comparison between MLM, FEM and analytical result

As shown in Figure 5-8, the final result of MLM is greatly improved from the initial computation after three adaptive computations. Both FEM and MLM agree very well in the final results but both are slightly higher than the analytical solution. The discrepancy is somewhat expected because the analytical solution assumes the elastic body has an infinite depth in the *x*-direction, while the numerical solutions base on a finite dimension.

Example 5.3: Contact of a snap-fit

Snap-fits are commonly used for locking, attachment or part assembly. In practical application, sharp edges are often used in the snap-fit mechanism to provide positive retention. In addition, a snap-fit must have a large retention force but a small insertion force. Since these forces are the result of the contact between components of snap-fit, contact analyses are very important for designing a snap-fit mechanism. We

demonstrate here the use of MLM to analyze the contact forces of a snap-fit, and to compare the results against those computed by ANSYS.

A typical snap-fit geometry is shown in Figure 5-9. Without loss of generality the retention block (that is assumed to be un-deformable) moves horizontally from right to left (x_c is not fixed in this case). The cantilever-hook is clamped at the left end and deflected upon contact; the geometry and material parameters of the cantilever-hook along with the options used for ANSYS and MLM are given in Table 3.

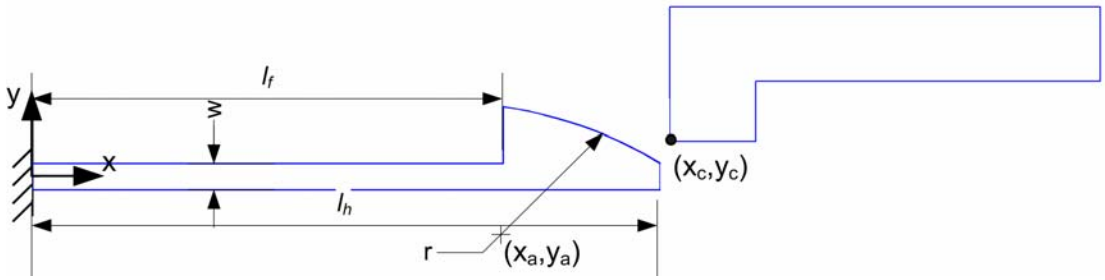


Figure 5-9 Geometry of a snap-fit mechanism

Table 5-3: Simulation parameters of snap-fit mechanism

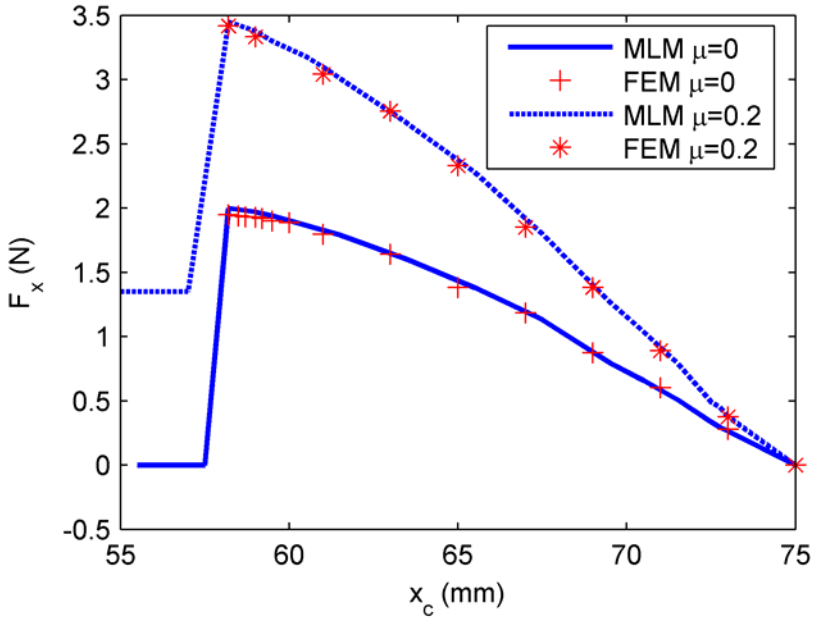
Parameters	**Values**	**Numerical 2D model**
Young's modulus (Pa)	2.62E9	Plane stress with a thickness of 10mm
Poisson's ratio	0.4	**ANSYS** with 3282 nodes
Thickness *w* (mm)	3.2	Element type: PLANE2, CONTACT175, and TARGET169
l_f (mm)	57	**MLM**
l_h (mm)	76	Number of nodes: 169 (initial)
Radius *r* (mm)	50	180~200 (after two adaptive computations)
x_a, y_a (mm)	49.9, -41.0	
y_c (mm)	2.6	

Figure 5-10 compares the contact forces computed using MLM against those obtained using commercial FEM package ANSYS for both frictionless ($\mu=0$) and

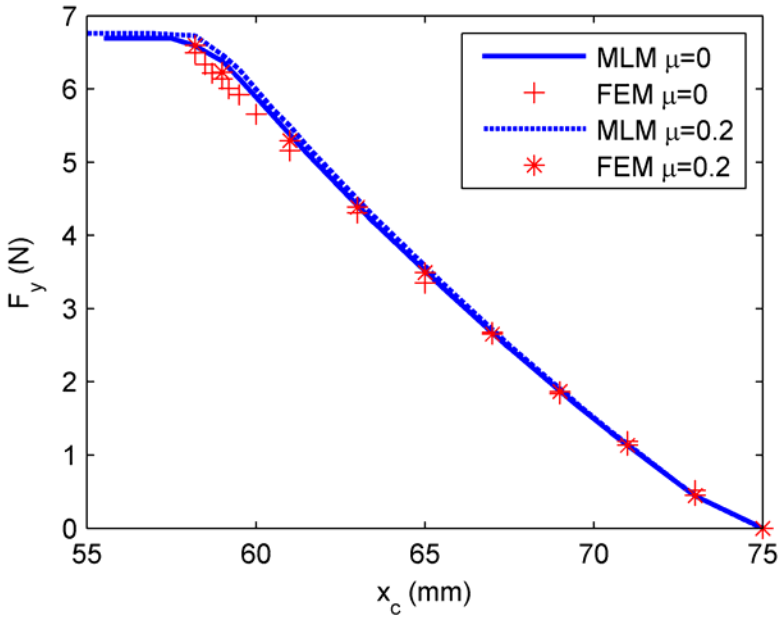
frictional ($\mu=0.2$) contacts. The MLM and FEM agree closely with each other up to the location where the edge of the retention block passes the tip of the jaw, beyond which the ANSYS computation breaks down due to the large distortion of elements. Unlike FEM, MLM is free from mesh distortion, and predicts the contact forces throughout the snap fitting process. It is interesting to note that the contact force in the y direction only increases slightly with friction, but the contact force in the x direction increases significantly. This result suggests that the cantilever-hook surface should be smooth in order to reduce the insertion force of a snap-fit.

Example 5.4: Contact simulation of needle insertion

Subcutaneous insertion of needles is one of the most common procedures employed in modern clinical practice. Applications of these procedures include the biopsy of deep-seated lesions, prostate brachytherapy, and neurosurgical probe insertion, which are usually performed without visual feedback from below the skin's surface. Maximum force and stresses generally occur at contact before penetration. As demonstrated in this example, the adaptive MLM can provide computationally efficient detailed information at the contact region between the surgical tool and tissues for applications in the medical surgery simulation. Specifically, we simulate here a needle contacting an elliptical elastic body. The material properties of the deformable body and the initial geometry and node distribution are shown in Figure 5-11. No special refinement has been made around the contact region for initial node distribution. The needle moves vertically downward from its initial position. The contact is computed from the tip of the needle at four locations starting from the location at 9.99E-3 and then increasing at an interval of 0.25E-3.



a) *x* direction force



b) *y* direction force

Figure 5-10 Contact forces

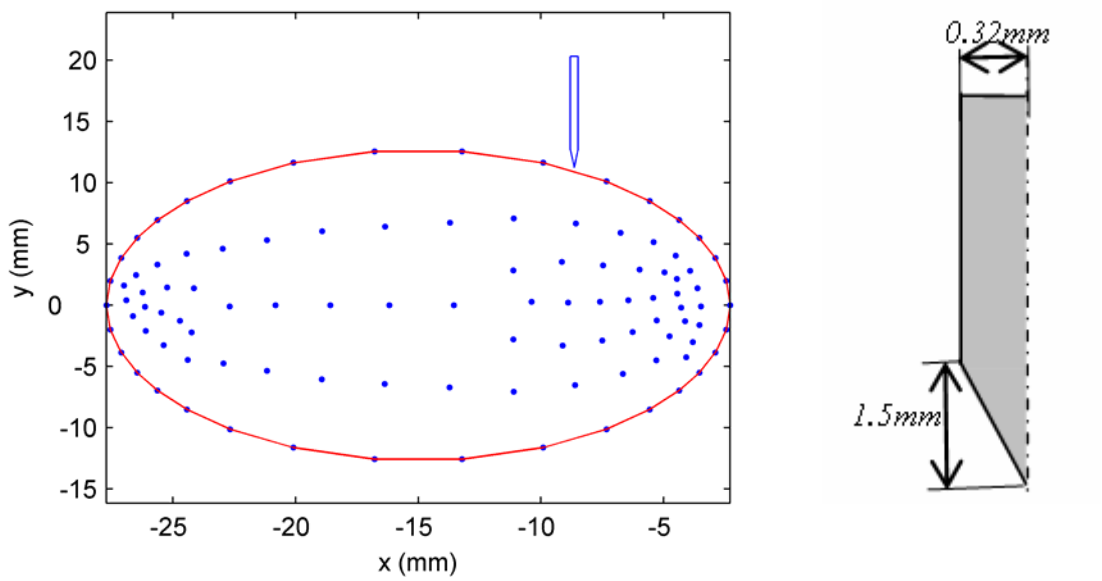


Figure 5-11 Initial geometry and node distribution

(Deformable body: Young's modulus E= 1E6 Pa; Poisson's ratio μ=0 .4)

At the initial or 1st contact position, four adaptive computations are performed. The converged results for the initial node distribution and the three subsequent adaptive computations are shown in Figure 5-12(a) to (d). Figure 5-12(a) shows that the computation with a small number of initial nodes cannot reflect the detailed deformation at the contact location. With more nodes used around the contact region, the small shape change can be seen at that location from Figure 5-12(d). The corresponding contact forces of the four computations at the initial position are shown in Figure 5-12(e). The convergence can be observed from the fact that the difference between the contact forces from two computations becomes smaller as the adaptive procedure continues.

Inheriting the nodes added from the 1st position, three additional adaptive computations are performed at the 2nd position. No significant improvement was observed between these computations, indicating that the node density is sufficiently

large. The deformed geometry and the node distributions at the other three positions (9.24E-3) are plotted in Figure 5-13(a)-16(c). The final results of the contact force at the four locations are listed in Table 4. The contact force increases as the needle moves downward.

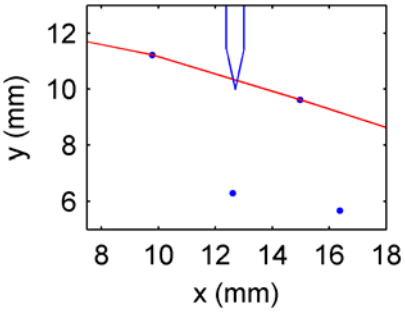
Figure 5-14 (a)-(d) shows the equivalent stress distribution around the contact region for each of the needle positions. As expected, the magnitude of stress increases as the needle moves from position 1 to 4, and its maximum occurs at the contact location. The stress information, which serves as the criterion for material failure in the theory of fracture mechanics, provides a means to judge when the penetration happens.

Table 5-4: Contact Force

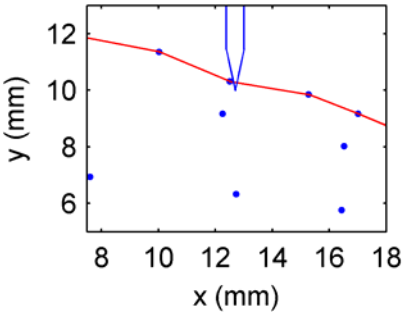
Location of needle tip(mm)	9.99	9.74	9.49	9.24
Contact force (N)	24.2	31.1	36.5	42.6

5.7 Summary

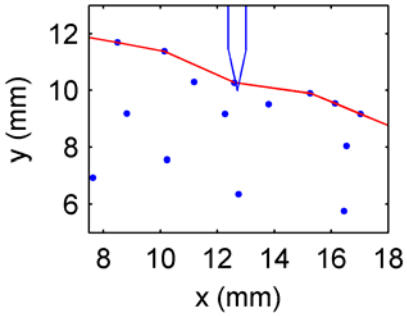
An adaptive MLM method for solving mechanical large deformation and contact problems has been presented. This method, utilizing sliding line and penalty methods for handling contact constraints, has been validated for two different situations; namely, large deformation and contact. Four practical examples have been illustrated. Simulation results show that the adaptive MLM algorithm can effectively identify regions of large computational errors, and can progressively add nodes accordingly; as demonstrated using intermediate results, the overall error is reduced as the adaptive procedure goes forward. These illustrative examples also demonstrate that the adaptive MLM has potential in robotic applications (such as surgical simulation and food processing) where the objects being handled are highly deformable.



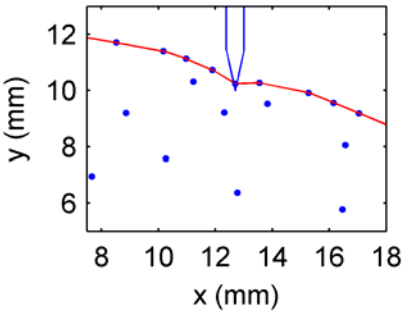
(a) Initial



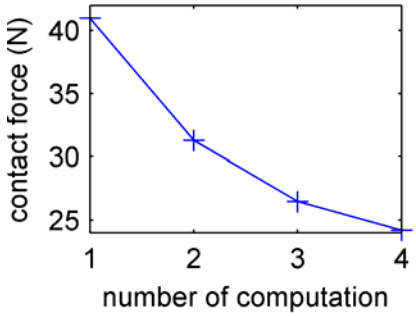
(b) After 1st computation



(c)After 2nd computation

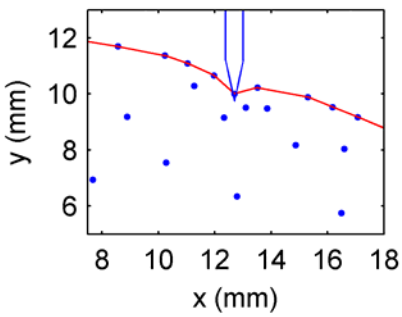


(d) After 3rd adaptive computation

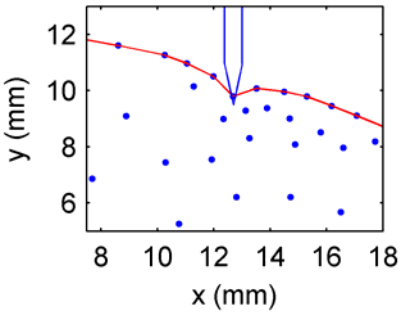


(e) Contact force

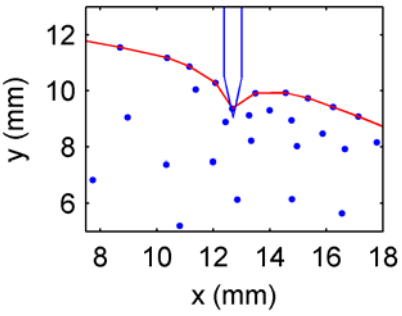
Figure 5-12 Result after each adaptive computation at the 1st position



(a) 2nd position

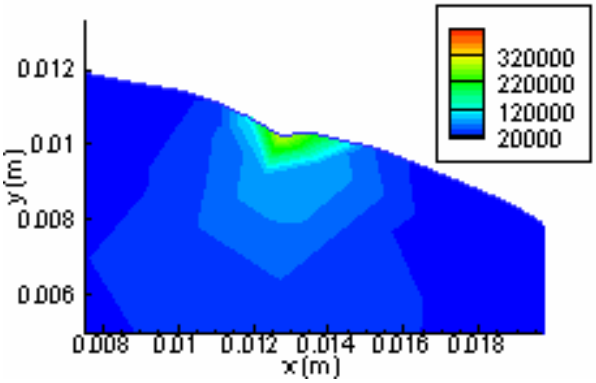


(b) 3rd position

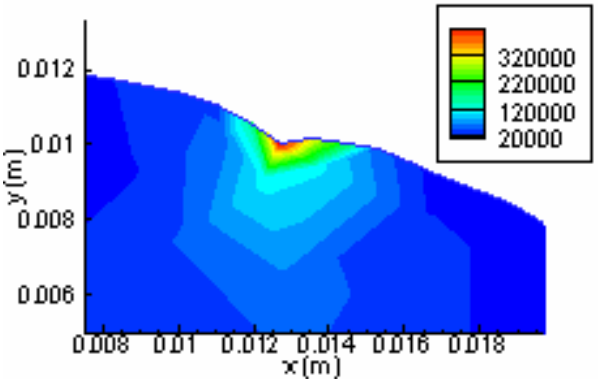


(c) 4th position

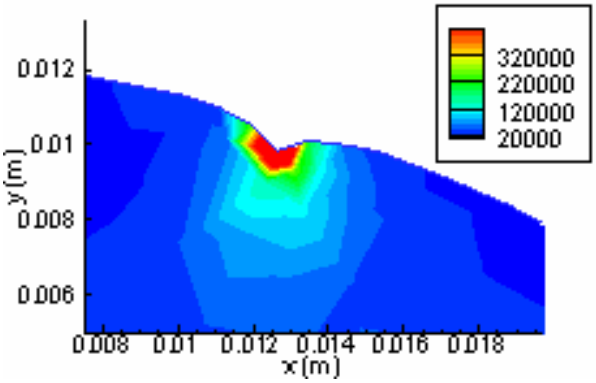
Figure 5-13 Results of MLM simulation



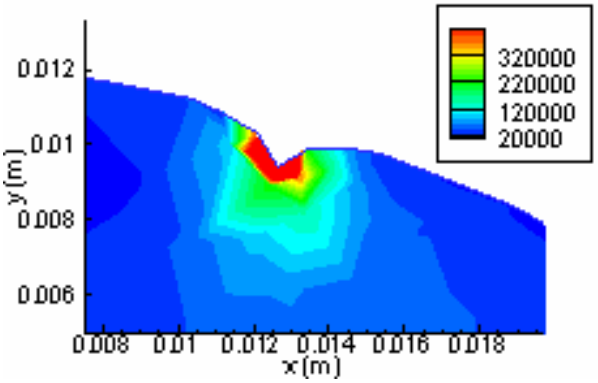
(a) 1st position



(b) 2nd position



(c) 3rd position



(d) 4th position

Figure 5-14 Equivalent stress distribution (N/m^2)

CHAPTER VI

ILLUSTRATION APPLICATIONS

6.1 Introduction

This chapter presents the application of the adaptive MLM for solving two types of engineering problems encountered in the development of a high-speed live-object handling system.

Transferring and handling live objects is needed in many industries. In the poultry industry, live birds must be transferred from cages to a killing line. In this process, birds are grasped and shackled. The task is currently handled by human workers at the poultry processing plants. An automation system is an ideal replacement for human workers, and is being developed at Georgia Tech. The system consists of subsystems that are able to automatically singulate, grasp and hang the bird mechanically [1, 5, 138]. An overall workflow of the system is given in Figure 6-1.

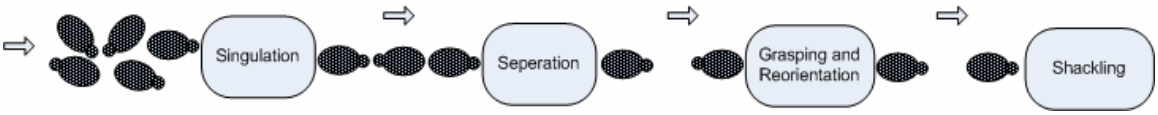


Figure 6-1 The overall workflow of live-object handling system

The operation of the system is briefly introduced as follows. The birds released from cages are unloaded onto the infeed conveyor where they are, in general, crowd randomly distributed. The birds are moved forward towards the singulator. They are separated and aligned into a single queue. Based on the bird's location on the singulation conveyor, minor adjustment for the distance between birds is made by changing the speed

of singulation conveyor so that the birds are loaded into the separation conveyor at a specified time interval. This enables the loading finger drum to accurately grasp and load the bird into the compartment running on the separation conveyor. The motion of the loading chain conveyor is synchronized with the motion of the moving flexible finger grasper so that once the broiler in the constraint compartment reaches the grasper, the grasper will be triggered to close and finish the grasping of the broiler.

After the broilers have been grasped by flexible finger grasper, they will be lifted from loading conveyor and transferred to the shackling machine where they will eventually be shackled. In order to simplify the subsequent process and reduce the difficulty of shackling the broiler, the broilers must be oriented in a particular direction before they are shackled. A re-orientation system that incorporates a real-time machine vision algorithm [139] has been developed to sense the orientation of the broiler and rotate the birds whenever they are facing wrong direction.

The remainder of this chapter is organized as follows:

1. In the first example, the MLM is applied to facilitate the design of a mechanical-magnetic actuator for manipulating the orientation of the live bird. The important design parameters of the actuator are determined based on the dynamic response of the actuator simulated using MLM.
2. In the second example, the MLM is applied to model the deflection of the flexible finger and to compute the contact force acting on the object by a flexible finger grasper.

6.2 Non-contact Live Bird Reorientation System

For high-throughput processing [22], re-orientation and alignment are often executed mechanically, which are often sources of wear and tear.

6.2.1 Experiment setup

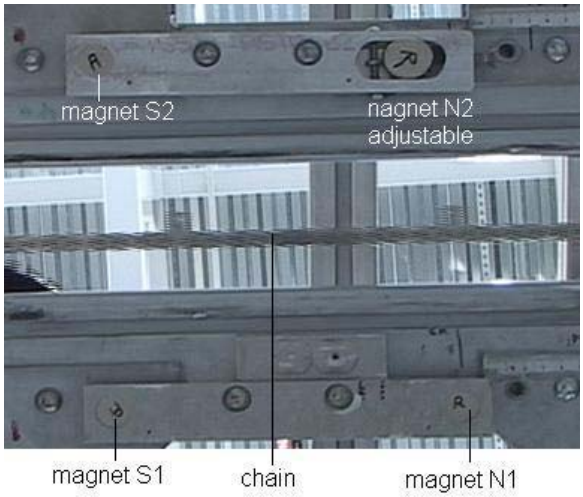
Figure 6-2 shows an overview of the re-orientation system. In this figure, a live bird is held by a pair of flexible finger graspers and is being transferred from the separation conveyor to the shackling machine. The re-orientation system consists of a vision camera, two fluorescence lamps, a vision algorithm and a magnetic actuator. To detect the orientation of the live bird, the camera takes a picture of the live bird when the bird passes under the two fluorescence lamps. The camera sends the picture to a computer. Based on the captured image, the real-time machine vision algorithm [139] computes the orientation of bird and triggers the magnetic actuator to rotate the grasper if the bird is facing wrong direction.

Figure 6-3(a) shows non-contact magnetic actuators used in the automated live-bird transfer system to orient birds. This high coercive permanent magnet system is used as an example to illustrate the application of MLM for analyzing the dynamics of a magnetic actuator.

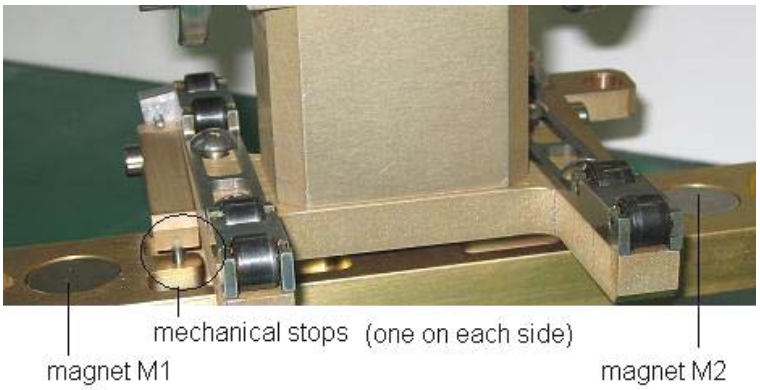
The magnetic actuating system consists of six permanent magnets. As shown in Figure 6-3 (b), four magnets are fixed on the underside of the track, and two magnets are embedded in the moving grasper driven by a motorized chain at a specified velocity $v(t)$ as shown in Figure 6-3 (c). The operational principle of the magnetic actuating system is illustrated in Figure 6-4, where the shading of the magnets indicates their polarities. The

grasper is initially locked in its horizontal position but is allowed to translate toward the pair of magnets (N_1 and N_2) positioned with an offset D_o. When a change in the grasper orientation must be made, the grasper is unlocked electromechanically as it passes a designated position. Repulsion causes the grasper to rotate (in $-\theta$ direction) to a specified orientation (defined by the pair of alignment magnets) within an often very short cycle time while the grasper is continuously moving.

Figure 6-2 Reorientation actuator and machine vision system [139]



a) Fixed magnets



b) Magnets on grasper

Figure 6-3 Reorientation and alignment in live-bird transfer system [124]

6.3.1 Dynamic model

The dynamic response of the actuator is determined by a number of design parameters including the magnetic size, the air gap length, and the offset distance D_0. The the magnetic size and the air gap length are the key parameters that characterize the magnetic force of the actuator. In general, a larger magnet size and a smaller air gap can provide a larger magnetic force. However, the magnetic size can not be too large since

the space is limited for actuator. In addition, the air gap can not be too small as well due to the vibration of the grasper. A good estimation for these two parameters can be obtained from those mechanical constraints. Once the first two parameters are decided, the last parameter, the offset distance D_0, becomes very important since this parameter can significantly affect the final dynamic response of actuator.

As shown in Figure 6-3(c), one of the two actuating magnets is made adjustable so that the offset D_o can be varied to achieve the optimal response of the actuator. It is of interest to determine

1. The effects of the offset D_o on the trajectory of the moving magnets and the torque on the grasper.
2. The optimal location to unlock to initiate the rotation.
3. The rotational trajectory $\theta(t)$, which will be experimentally verified.

In order to investigate these effects, the equations for simulating the dynamic response of the actuator are derived. To simplify the formulation, the coordinate system XY is defined at the center of the rotating grasper as shown in Figure 6-4(b). Thus, the fixed magnets move with a constant speed v(t) in the negative Y direction. The coordinates of the two magnets N_1 and M_1 are respectively given by

$$\mathbf{x}_{N_1} = \begin{bmatrix} -R \\ y_1 - vt \end{bmatrix} \tag{6-1}$$

and

$$\mathbf{x}_{M_1} = -R\begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix} \tag{6-2}$$

where y_1 is initial location of magnet N_1 when the grasper is unlocked. The actuating torque as a result of the repulsion between M_1 and N_1 has the form:

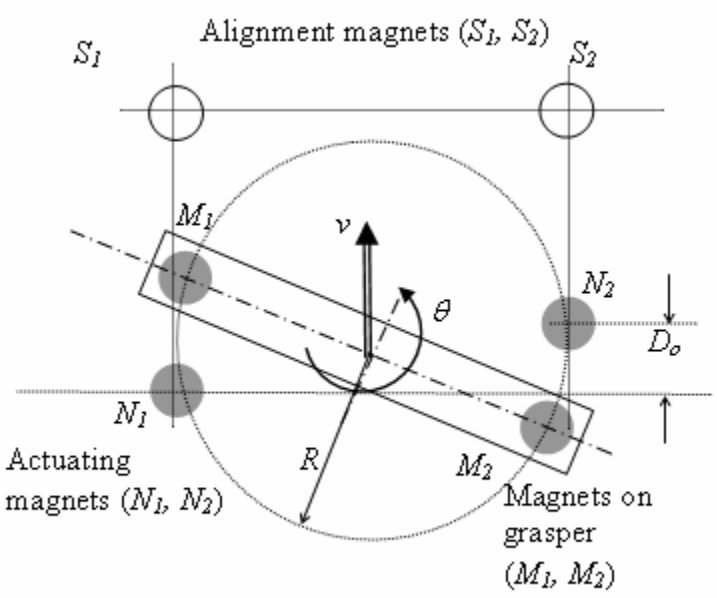
$$T_1 = f(\|\mathbf{D}_{m_1}\|)\left(\mathbf{x}_{M_1} \times \mathbf{D}_{m_1}\right) / \|\mathbf{x}_{M_1} \times \mathbf{D}_{m_1}\| \tag{6-3}$$

where $\mathbf{D}_{m_1} = \mathbf{x}_{M_1} - \mathbf{x}_{N_1}$. The torque T_2 due to N_2 and M_2 can be determined similarly. The equation of motion for solving the trajectory is given by

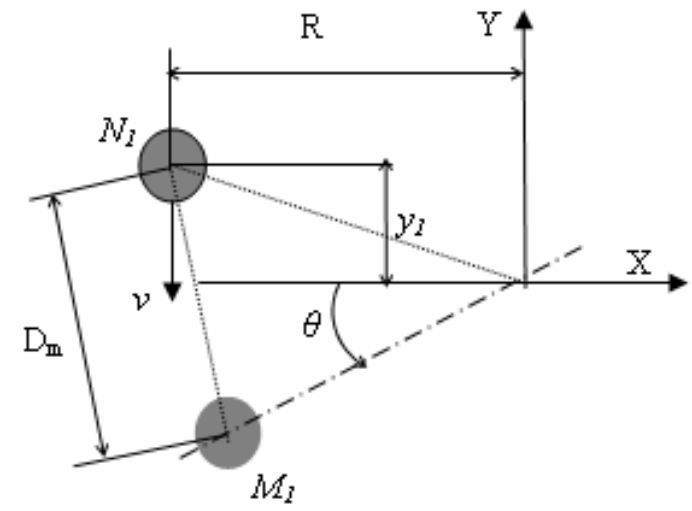
$$\ddot{\theta} + 2\xi\omega_n\dot{\theta} + \omega_n^2\theta = \left(T_1 + T_2\right) / I \tag{6-4}$$

where I is the rotating inertia; ξ is the damping ratio; and ω_n is the natural frequency. To solve for $\theta(t)$, we determine the force $f(\|\mathbf{D}_m\|)$ from the magnetic field using the meshless method.

The damping ratio and the moment of inertia of grasper are determined experimentally as shown in Figure 6-5. First, the grasper is tied to a fixed structure from the both sides with a pair spring whose stiffness is known. Next, the grasper is turned a small angle from its balanced location and released suddenly such that the structure vibrates freely. Finally, the vibration of the structure is recorded to a video tape which is digitized later. Since the friction is small, this whole system is an underdamped second order system. The natural frequency of the system can be obtained first from the vibration. The moment of inertia and the damping ratio are computed using the spring stiffness and the natural frequency.



a) Illustrating schematics



b) Coordinate system

Figure 6-4 Re-orientation and alignment actuating system

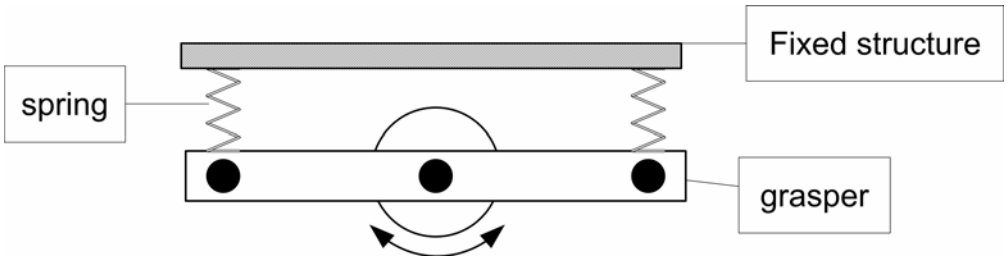


Figure 6-5 The experimental setup used to determine the parameters of the system

The values of the parameters used in the simulation are based on the live-bird transfer system shown in Figure 6-4(a) so that computed results can be validated experimentally. The parameter values are given in Table 6-1:

Table 6-1: Nominal values used in the MLM simulation

Magnets			
material	$\mu_0 M$ (Tesla)	diameter (mm)	thickness (mm)
Neodymium grade N38	1.24	25	12.5
Actuator configuration			
D_o (mm)	2R (mm)	g (Gap between rotating and the fixed magnets) (mm)	
25	175	6.25	
Grasper			
I (kg-m)	ξ	ω_n (radius/sec)	v (m/sec)
0.0316	0.024	7.42	0.45
Meshless: (weak form) 41×41points;			

6.3.2 Effect of mass variation

The mass of grasper has a significant effect on the dynamic response of the system since larger mass results in longer system response time as shown in Figure 6-6. Several considerations have been applied to reduce the weight of the grasper, such as

choosing a lower-density material for the bracket of the grasper and making the grasper bracket a hollow structure.

6.3.3 Effect of D_o

As shown in Figure 6-7, the offset D_o has a significant effect on the shape, peak value and timing of the rotating torque. The optimal toque is chosen based on the following two criteria:

1. The overall torque must be large enough to enable the system to rotate fast enough to satisfy the short cycle time requirement of the system.
2. The motion generated by the actuator must be smooth to reduce the reaction of the live birds.

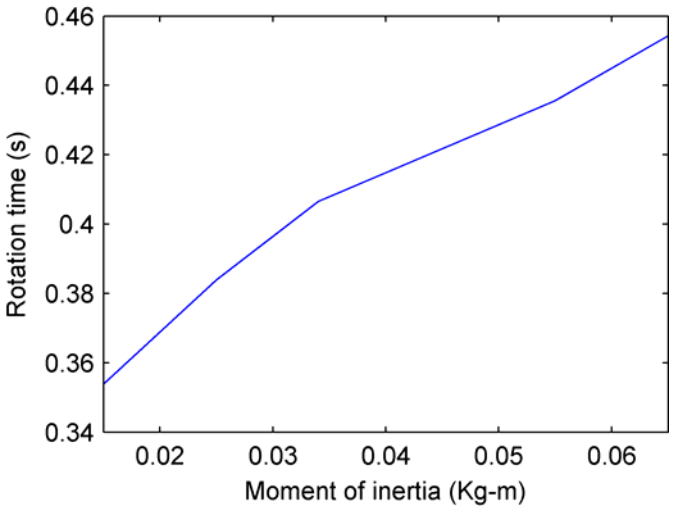


Figure 6-6 Effect of inertia

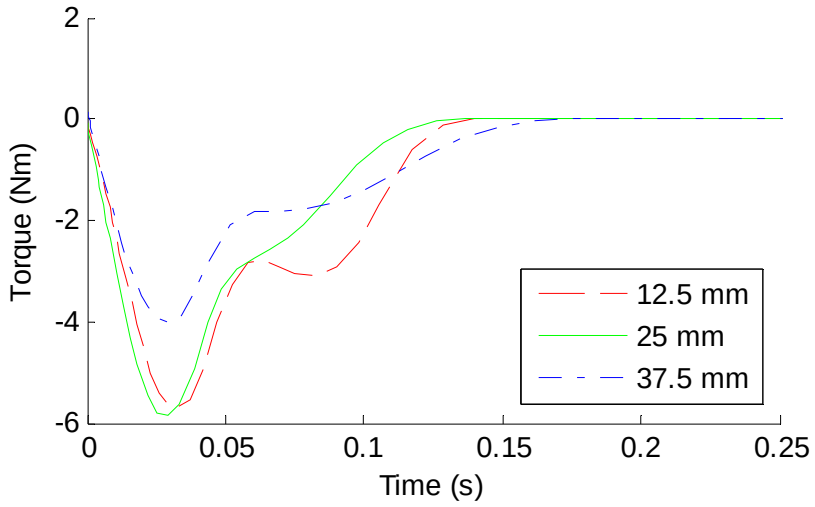


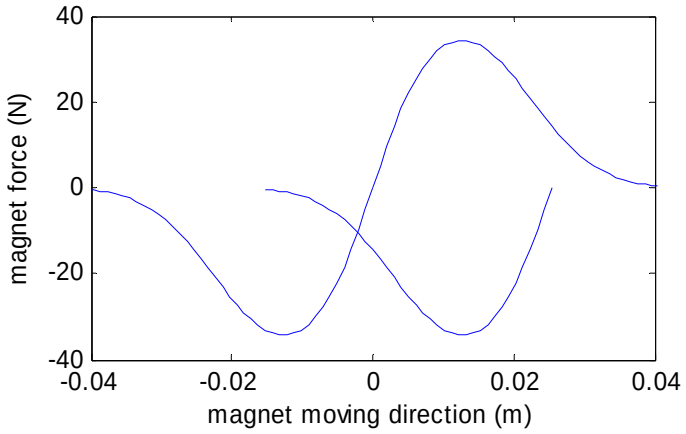
Figure 6-7 Effect of offset D_o on rotating torque

It can be observed from Figure 6-7 that the maximum magnitude of the toque increases slightly as the D_o increases from *12.5*mm to *25*mm. However, further increasing D_o up to *37.5*mm reduces the toque significantly because of the enlarged distance between the magnets. Hence, the D_o can not be too large. Although the maximum magnitudes of torque for the first two cases D_o=*12.5*mm and *25*mm are very close, the torque curve for D_o=*25*mm is much smoother at the end of the rotation. Thus, the optimal offset is found to be D_o=*25*mm (1 inch), which has been used in all subsequent simulations.

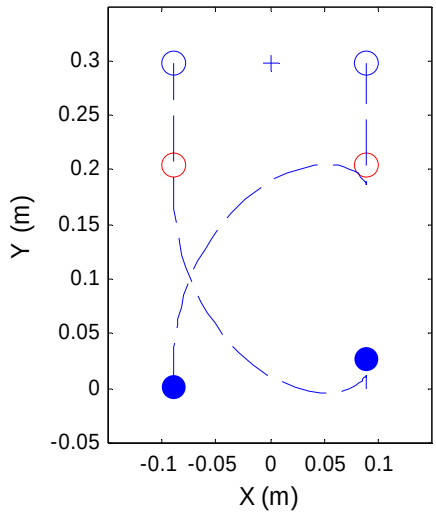
6.3.4 Determine the unlock location

To determine the optimal location to initiate the rotation, the pair of repulsive forces between N_i and M_i (i=1,2) is plotted quasi-statically as a function of displacement in Figure 6-8(a). Initially, M_2 approaches N_2 which results in a large negative force on the grasper. Unlocking at this moment would require overcoming a significantly large static friction. However, as M_1 is moving into the range of N_1, the net torque acting on

the grasper gradually reduces, passes through zero, and becomes positive. The optimal instant to unlock for rotation is when the two opposing forces are equal, and thus the static friction is a minimum. From Figure 6-8(a), this location is -2mm from N_1, with which the simulated trajectory of the two rotating magnets is graphed in Figure 6-8(b).



a) Individual static force



(b) Magnetic trajectory

Figure 6-8 Effect of repulsion magnets on rotation

To verify the computed trajectory of the grasper, the motion of the rotating magnets is digitally recorded using a 3-CCD video camera. As compared in Figure 6-9, the experimentally obtained $\theta(t)$ agrees well with the values computed from Equation (6-4) with the magnetic force computed using the meshless method.

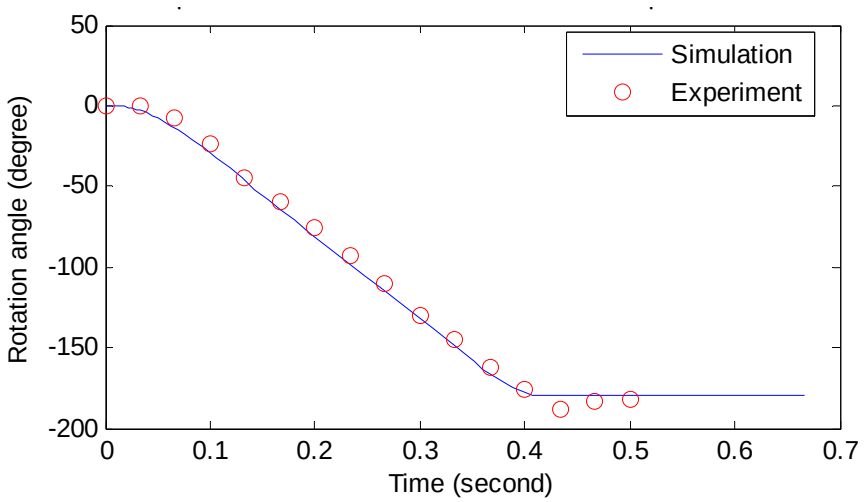


Figure 6-9 Comparison between simulation and experiment

6.3 Contact with Large Deformation due to Flexible Finger

In this section, the meshless method computational model is utilized to predict the deflection of the flexible finger due to contact. Figure 6-10 schematically shows a typical scenario that happened in the process of singulation, loading or grasping. Since the grasper relies on contact between the flexible fingers and the live bird to manipulate the live bird, analyzing the contact is very important for a successful grasping manipulation.

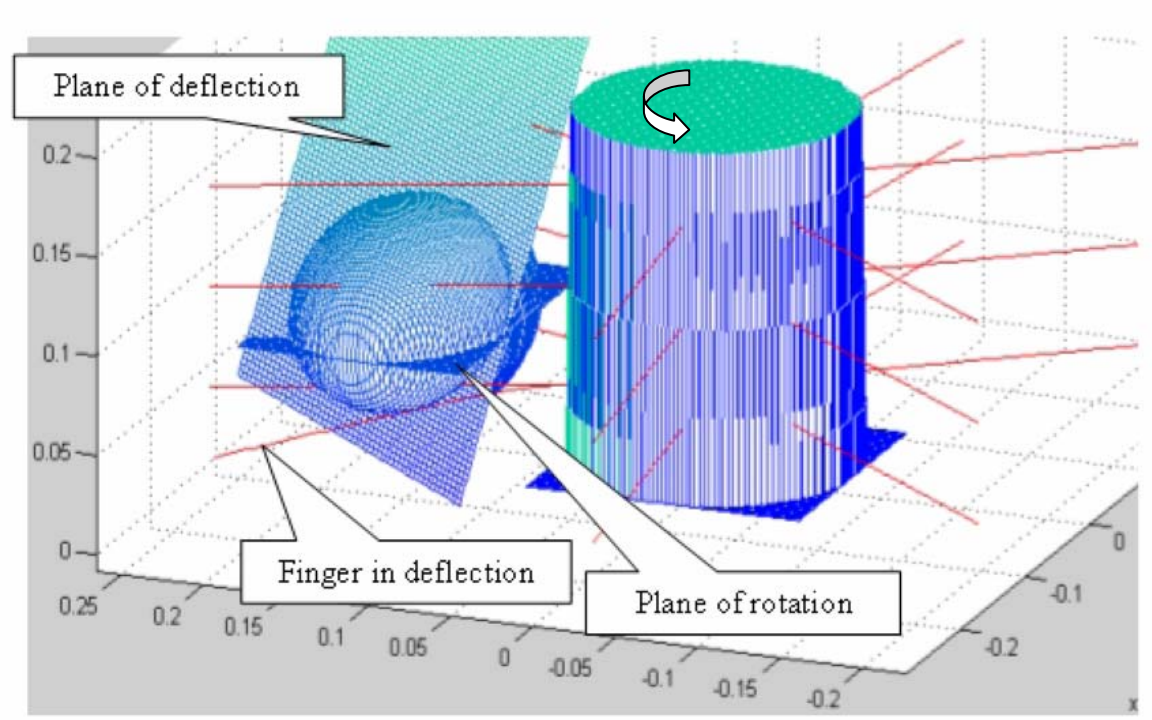


Figure 6-10 Cross section of the body of the live object [140]

Figure 6-12 shows the 3D geometry model of a 4in flexible finger built in the CAD package Solidworks. The dimensions of the finger can be found in [141]. [6, 141] simplified the flexible finger to a 1D nonlinear beam and studied the contact between the flexible fingers and the body of a live object. The investigation shows that the computational results using the beam model match the experimental result very well when the finger is long and the contact point is far away from the clamped end of the finger. However the result becomes less accurate when the contact point is close to the clamped end of the finger. This is mainly due to two reasons:

1. The beam model is derived using the assumption that the length-to-thickness ratio of beam is large. This assumption is violated when the finger is short or the force is applied near the root of the finger.

2. The beam model ignores the finger's shape. The true contact point can not be located accurately without considering the thickness of the finger. This is especially true when the contact point is close to the root of the finger where the thickness is much larger.

In this research, a 2D finger model is developed using the plane stress assumption based on the following observations:

1. The stress along the horizontal direction of the finger is assumed to be sufficiently small that the deflection can be modeled by the plane stress model.
2. Since the vertical dimension of the finger is much smaller than the horizontal dimension of the finger, the finger deflects mostly in the vertical plane.

Compared with the general 3D computational model, the 2D model lacks the ability to simulate more complicated situations such as twist. However, this 2D model improves the accuracy of contact simulation by considering the effect of the finger's thickness. In addition, the result obtained by 2D model is rather accurate when the force is applied in the plane of the finger's vertical cross section. Besides, the 2D model is computationally more efficient than the 3D model. Past research has developed a method to design the finger configuration such that the 3D grasping situation can be approximated by 2D [140]. The basic idea is shown in Figure 6-10. For each given instance during the process of grasping, the vertical plane of the finger intersects with the body of the live object (ellipsoid shape) to form a 2D ellipse. This 2D ellipse is used as the shape of the live object. Since the finger's contact surface has been chosen to be closely aligned with the tangential plane of ellipsoid when designing the grasper, the twist angle for the finger is neglected here. Past experimental study [141] that measured

the contact force between the finger and an aluminum ellipse provides a good means to validate the developed 2D model. The experimental setup is shown in Figure 6-11.

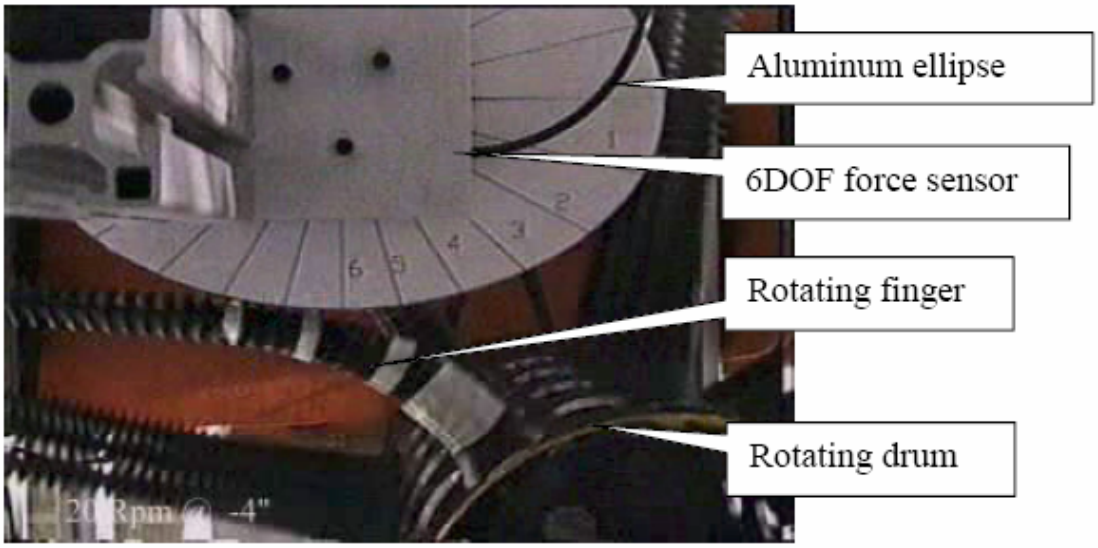


Figure 6-11 Experimental setups for measuring the contact force [140]

Three steps have been made to apply the meshless method to contact simulation of the flexible finger:

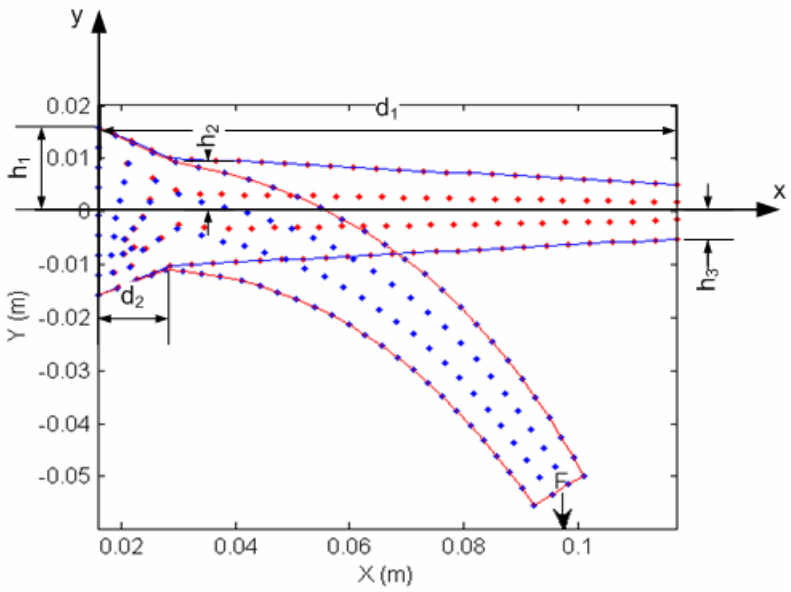
1. A 2D plane stress finger model is constructed using the 3D FEM computational result. The 2D model reduces the computational cost for subsequent contact computation significantly.
2. The reduced 2D model is validated by comparing the results against experimental results given in [141] and the FEM 3D results.
3. The reduced 2D model is applied to the contact computation between the flexible finger and the body of live object.

6.3.1 Construction of a 2D computational model for flexible finger

The finger's cross section is used to simplify the full geometry to 2D. The small geometry details on the finger's surface, such as finger ribs, are ignored because our preliminary study shows that these geometry details have contributed little to the final deflection shape of the finger while they increase the computational time. Figure 6-12b shows the 2D simplified meshless geometry model and one of computational results.

a) CAD model of flexible finger

Figure 6-12 Finger geometry



b) 2D meshless model

Figure 6-12 Continued

Table 6-2: simulation parameters

Parameters	Values	Parameters	Values
Applied Force (N, pound)	22.241 (5)	Finger length d_1 (m)	0.1016
Youngs modulus (Pa)	9E6	h_1 (m)	0.01588
Poisson's Ration	0.4	h_2 (m)	0.01016
d_2 (m)	0.0127	h_3 (m)	0.005189

An important design parameter for the plane stress model is the thickness of the 2D model. The plane stress model assumes the thickness of the model is uniform. However, the thickness of the actual finger is non-uniform. Rather than simply assign a value as the equivalent thickness, we use a procedure described below to determine an optimal equivalent thickness.

First, the deflection of the finger is computed using the geometry shown in Figure 6-12(b). One end of the finger is clamped and a force is applied vertically downward at the other end of finger. The magnitude of the force and material parameters are given in Table 6-2. Given the average finger thickness, we compute the deflections of the finger for 21 different thicknesses increasing from 13.5mm to 15.5mm with an interval of 0.1mm. These deflection results are shown in Figure 6-13.

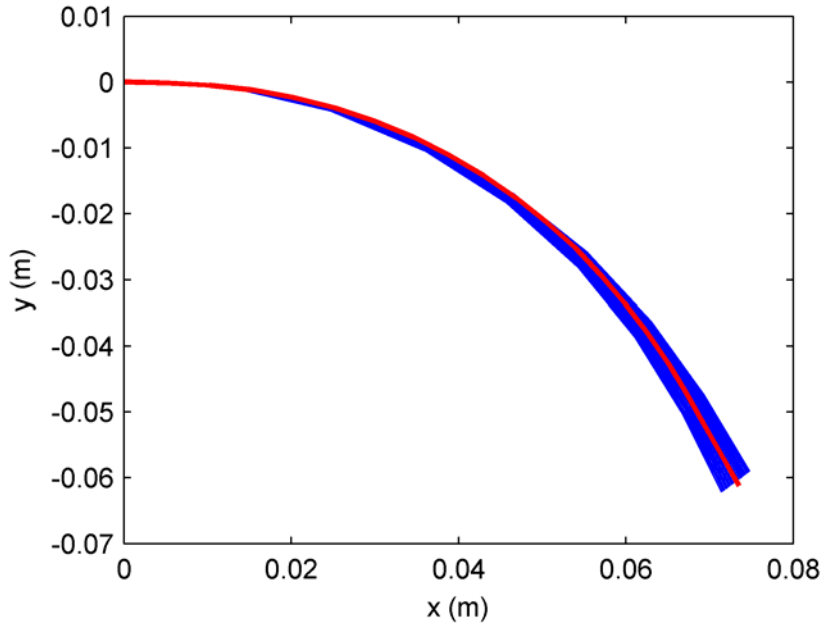


Figure 6-13 deflection of the finger for different thicknesses

(- FEM result, - MLM result)

Then, we repeat the computation using the 3D true geometry model and the commercial FEM package ANSYS. While the material properties and boundary conditions in the FEM computation are the same as these used in the meshless computation, the geometry model is imported from the one built in the CAD package Solidworks as shown in Figure 6-12a. To ensure the accuracy of the result for large

deformation, we choose the 3D 10-Node solid187 element. This element is a quadratic element and has good performance for handling highly irregular geometries such as those imported from a CAD system. The 3D deflected shape of the finger computed by ANSYS is shown in Figure 6-14.

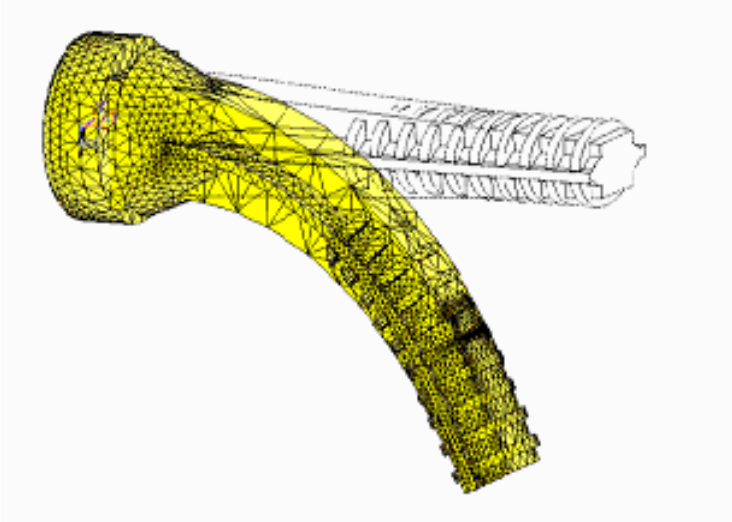


Figure 6-14 3D true-geometry finite element result

The error of the 2D meshless model for the different thicknesses can be computed as follows:

$$D_{nor} = \sum_{i=1}^{n} (d_i)^2 \tag{6-5}$$

where D_{nor} is the overall shape error for a thickness; d_i is the distance measured vertically from the i^{th} meshless node to the deflection curve obtained by FEM; n is the total number of meshless nodes. The change in the overall error of the meshless results with respect to the different thicknesses of the 2D model is shown in Figure 6-15.

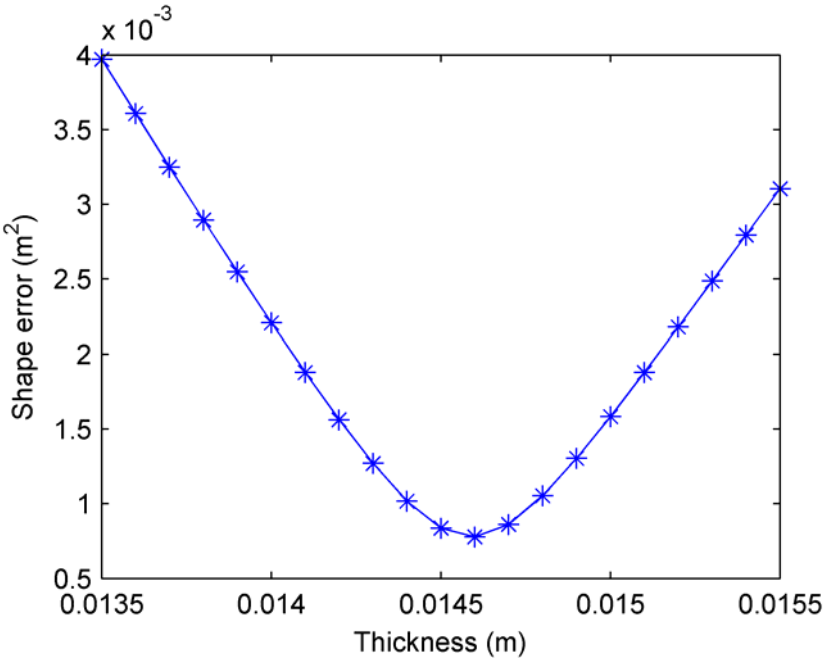


Figure 6-15 Shape error for different thickness

Based on the result shown in Figure 6-15, it is easy to determine that the optimal equivalent thickness is 0.0146m which is the location of the smallest error.

6.3.2 Validation of the 2D meshless finger model with experimental result

To further validate the 2D meshless model and the FEM model, the computational results of FEM and MLM are compared to the experimental result. Figure 6-16 shows the experimental setups. The comparison of the results among the MLM, FEM and experimental data is shown in Figure 6-17.

Figure 6-16 experimental setups [141]

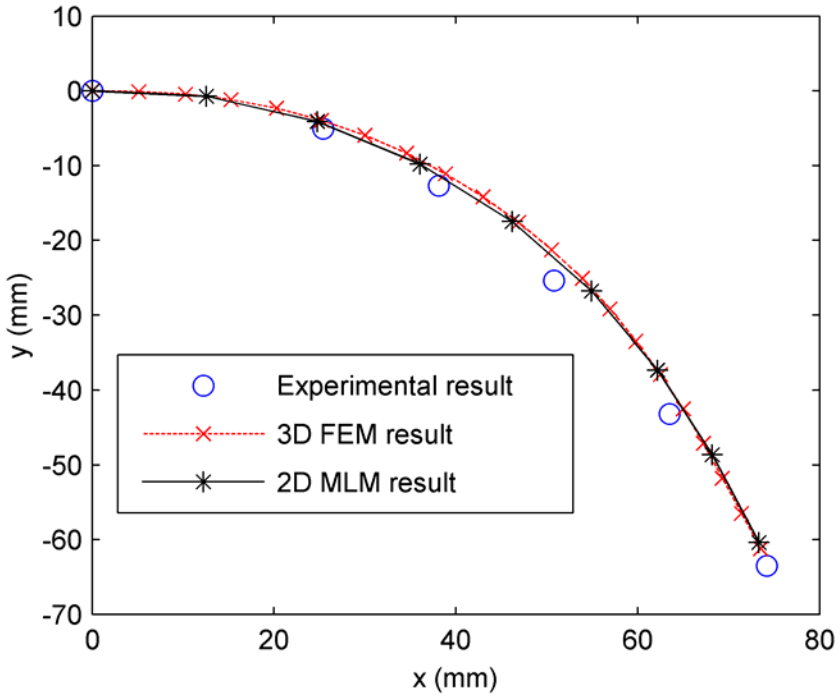


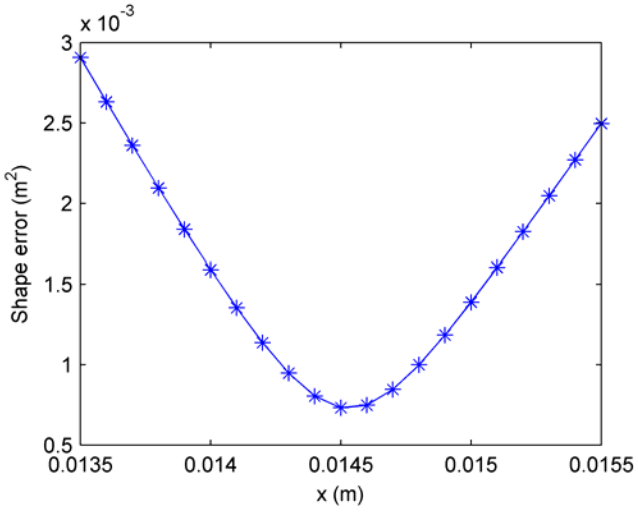
Figure 6-17 Comparison of finite element result, meshless result and experimental result

As observed in Figure 6-17, the 2D MLM result matches the 3D FEM result very well. The computational results from both of the methods also agree closely with the experimental result, which validates the computational models.

6.3.3 Equivalent thickness Function

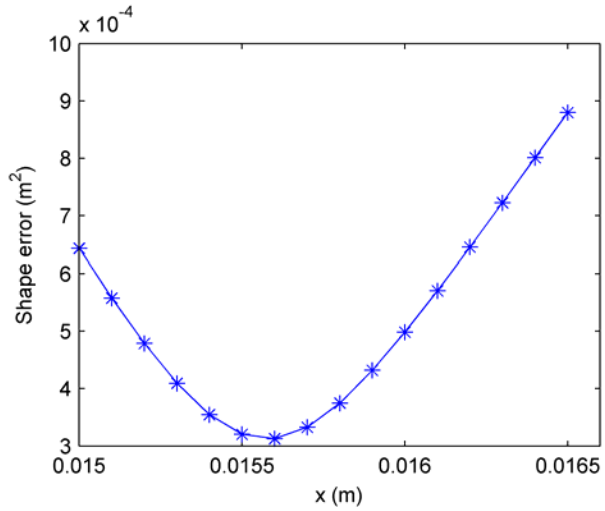
We demonstrate in Section 6.3.1 how to obtain the equivalent thickness for the 101.6mm (4inch) finger. Since the actual thickness of the finger is non-uniform, a fixed equivalent thickness will result in a large numerical error if the force is applied at different locations. Thus, we extend the equivalent thickness to be a function of the locations at which the force is applied.

To obtain the equivalent thickness function, the computation in Section 6.3.1 is repeated for three other lengths of finger: 76.2mm (3inch), 50.6mm (2inch) and 25.3mm (1inch) respectively. The shape of the error plots for these cases is given in Figure 6-18.

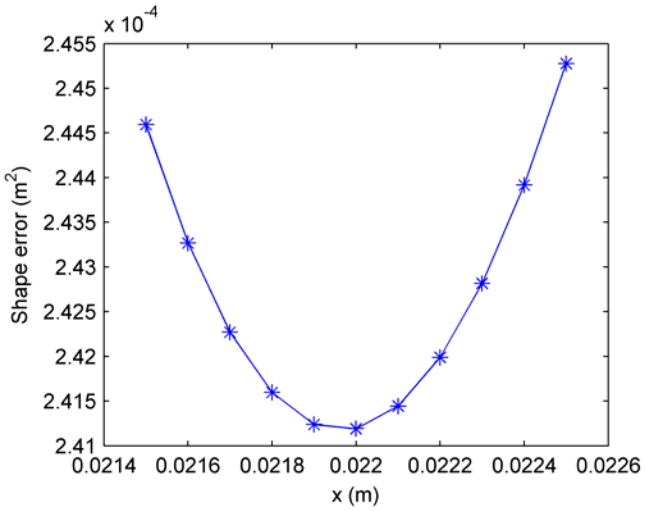


a) 76.2mm (3inch) finger

Figure 6-18 Shape error for different finger length



b) 50.6mm (2inch) finger



c) 25.3mm (1inch) finger

Figure 6-18 continued

Table 6-3 gives the optimal equivalent thickness for 4 different lengths of finger. The change of equivalent thickness is plotted in Figure 6-19. We observe that the equivalent thickness increases as the location where the force is applied becomes closer to the root of the finger. This is consistent with the fact that the thickness of the finger is larger at the root region.

Table 6-3: equivalent thickness for different finger lengths

Finger length, m (inch)	0.1016 (4)	0.0762 (3)	0.0528 (2)	0.0254 (1)
Equivalent thickness, m	0.0146	0.0146	0.0156	0.022

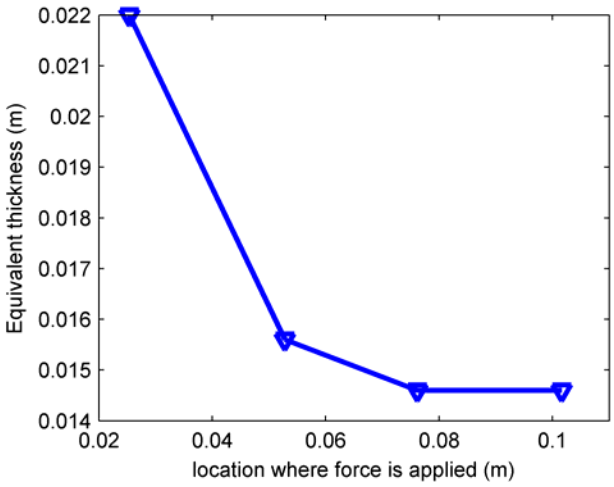


Figure 6-19 Equivalent thickness function

6.3.4 Large deformation flexible finger contact using meshless method

Once the equivalent thickness function is obtained, the 2D finger model is applied to study the contact between flexible finger and the object to be grasped. In this section, we compare the contact computation result with the experimental result given in [140, 141].

The experiment setup to measure the contact force between the flexible finger and the elliptical aluminum disk is shown in Figure 6-11. One end of the finger is fixed to a rotating drum. As the elliptical disk moves horizontally from right to left, the finger rotates counter clockwise and contacts with the disk. The contact force is measured as the finger rotates. The discretized meshless model is shown in Figure 6-20. The simulation parameters are listed in Table 6-4.

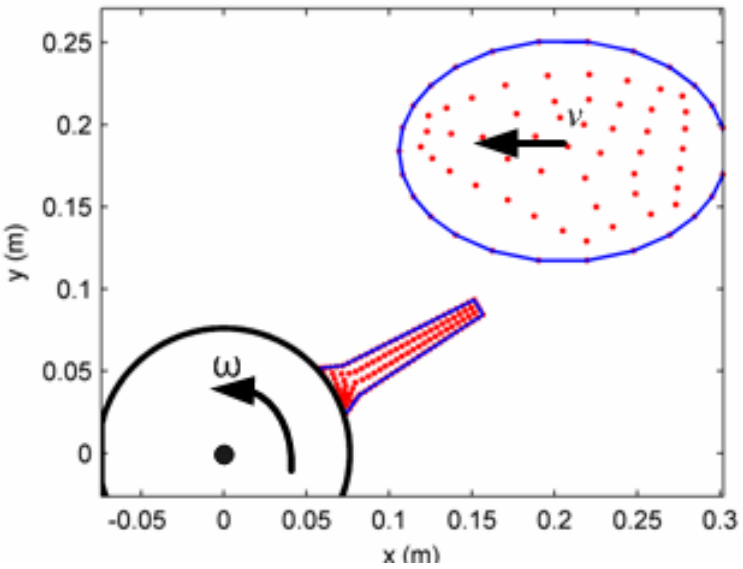


Figure 6-20 Computation model for meshless method

Table 6-4: Parameters for contact simulations

Parameters	Values
Initial location of the center of the bird	x_o=0.332 m; y_o=0.184 m
The initial angle of the finger	0^o
Ellipse's half width along the major axis	0.099 m
Ellipse's half width along the minor axis	0.067 m
Finger length	0.106 m (4 inch)
The radius of the drum	0.0762 m
The velocity of the ellipse (v)	0.508 m/s
The angular velocity of the finger (ω)	20 rpm

The contact is simulated using the algorithm developed in Chapter 5. Since the stiffness of the aluminum disk is much greater than that of the rubber finger, the aluminum disk is assumed rigid. Note that the effective thickness for each iteration is different due to the location change of the contact point, and it is computed by linear

interpolation from Table 6-3. Figure 6-23 shows the meshless computational result of the contact for the rotational angle of 80^{o}.

For comparison, the contact is also simulated using the 3D FEM. Figure 6-22 shows the discretized geometry and one of the computational results. The contact force is computed using the 2D ML model and the result is compared with those measured experimentally and with the 3D FEM contact model.

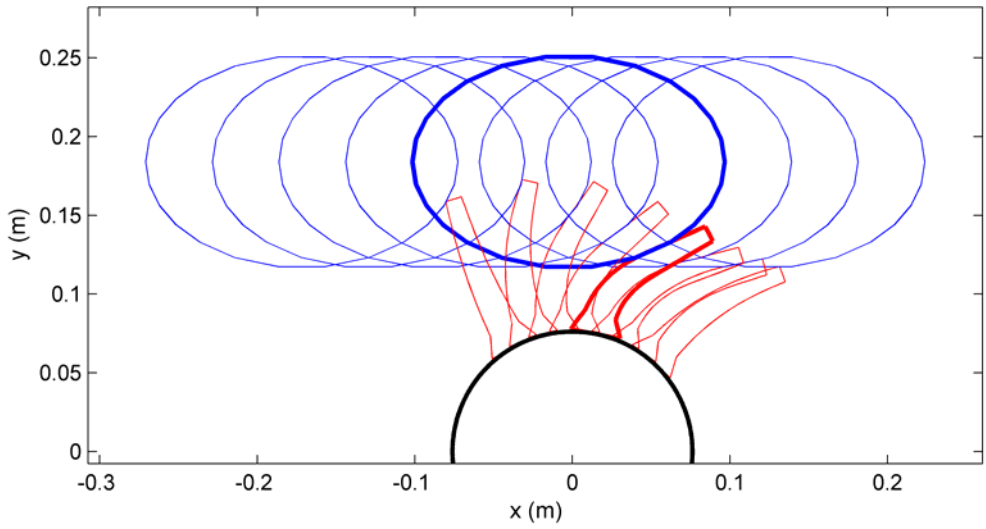


Figure 6-21 Contact results for a few chosen rotational angles of the finger

Figure 6-24 shows the comparison of the contact force obtained by FEM, MLM and experimental data. The contact force computed by the simplified 2D ML model matches the FEM result reasonably well.

To provide a better understanding, the effect of the object stiffness on the contact forces is simulated using three different material properties: aluminum, rubber and silicone. Their material properties are given in Table 6-5. In Figure 6-25, the stress

distribution in the body and finger at 80^{o} (where the maximum overall contact force occurs) is shown for different body materials.

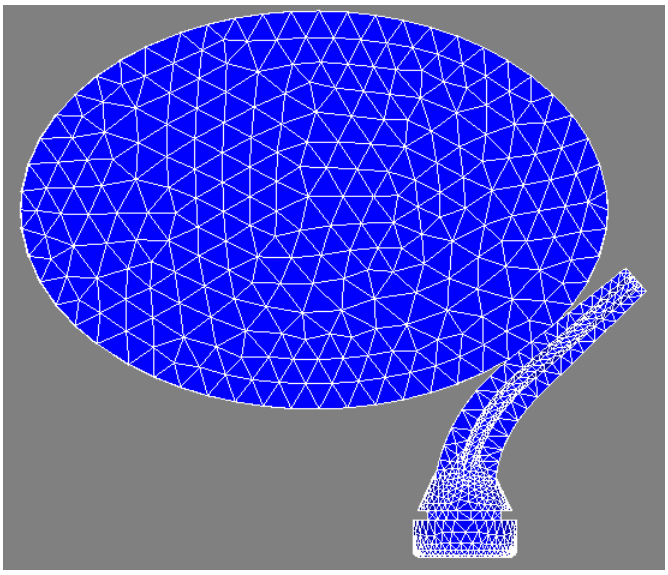
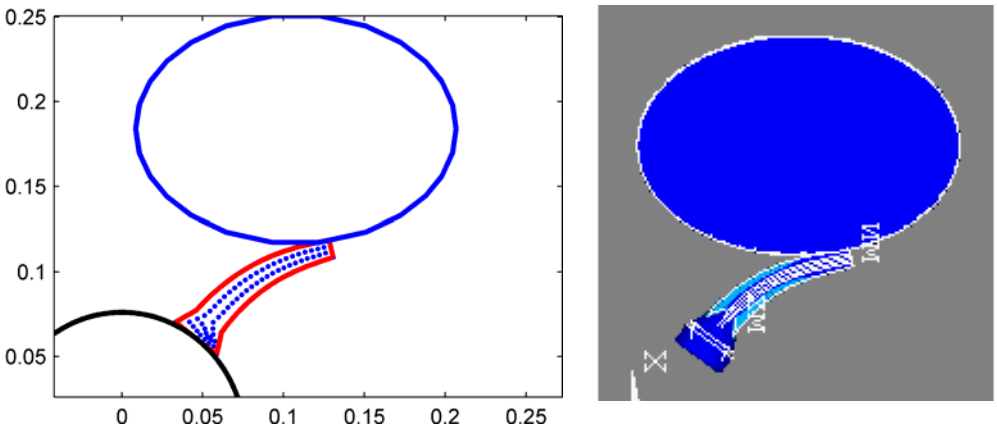
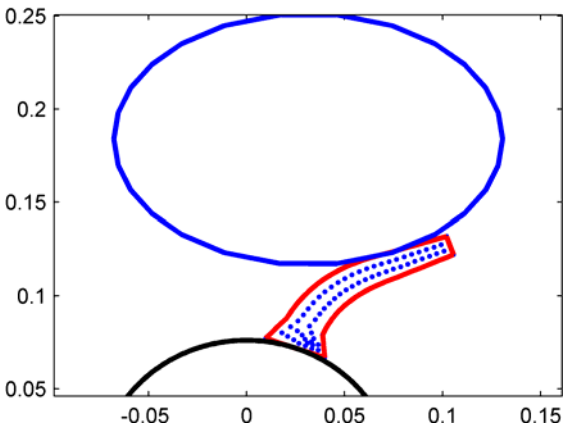


Figure 6-22 FEM mesh for contact computation

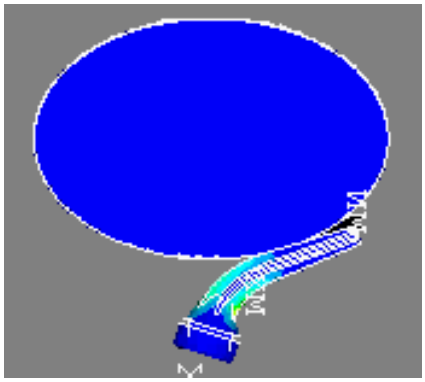


a) MLM contact result at 54^{o} b) FEM contact result at 54^{o}

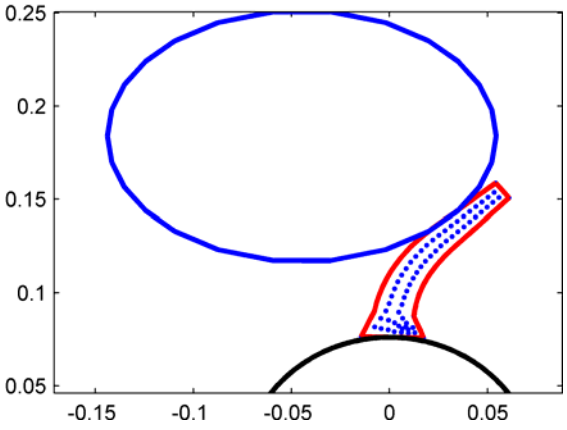
Figure 6-23 contact computation result of MLM and FEM



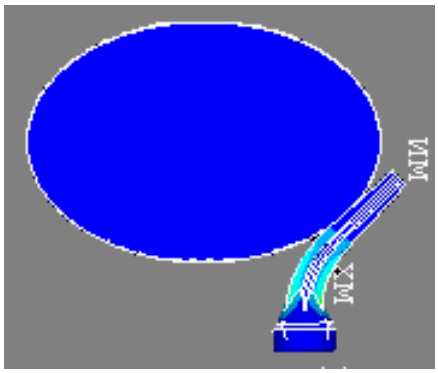
c) MLM contact result at 72°



d) FEM contact result at 72°

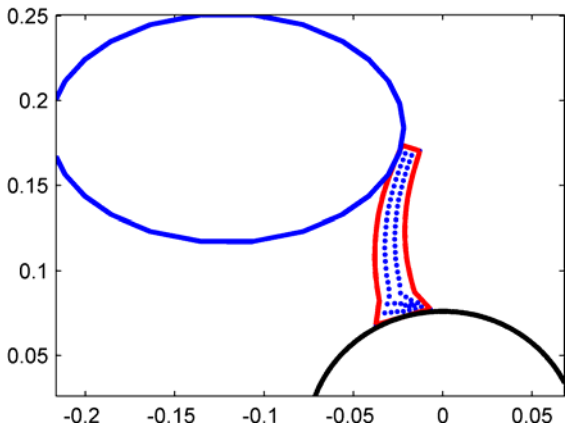


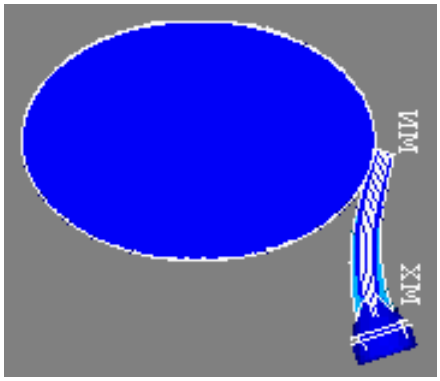
e) MLM contact result at 90°



f) FEM contact result at 90°

Figure 6-23 continued





g) MLM contact result at 108^{o} h) FEM contact result at 108^{o}

Figure 6-23 continued

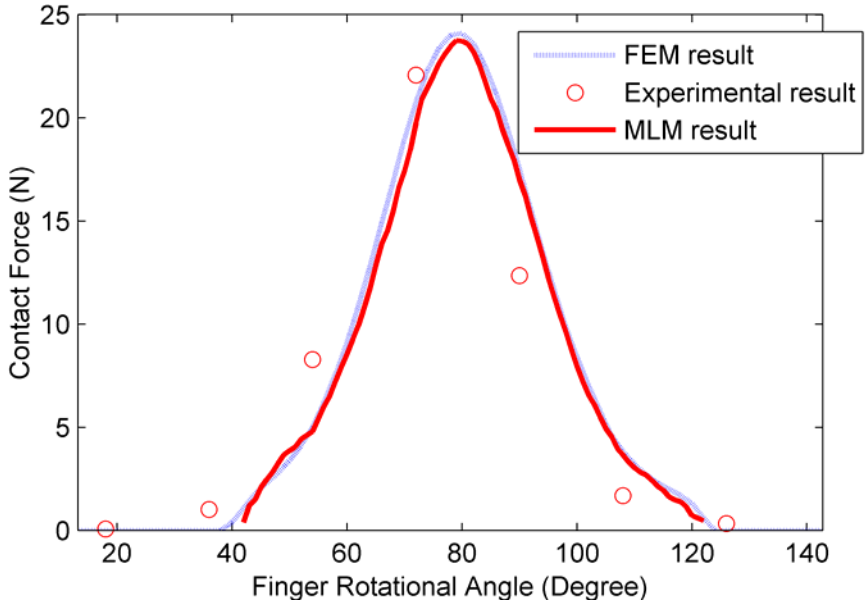
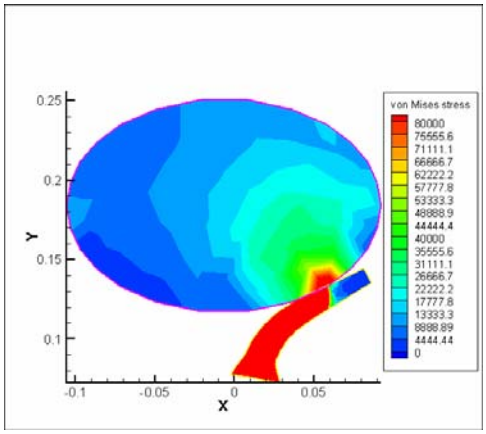


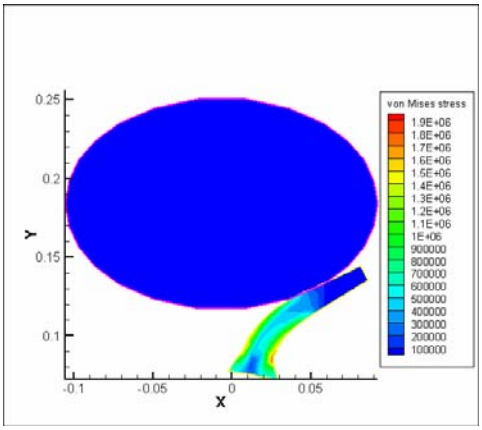
Figure 6-24 Contact force comparison

Table 6-5: Material properties

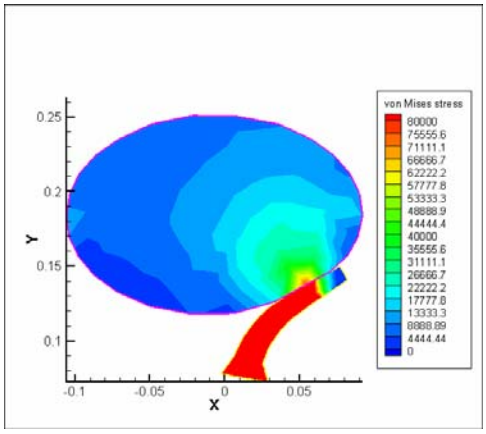
Materials	Aluminum	Rubber	Silicone
Young's modulus	*7.31 Gpa*	*9 Mpa*	*0.1 Mpa*
Poisson's ratio	*0.3*	*0.4*	*0.4*



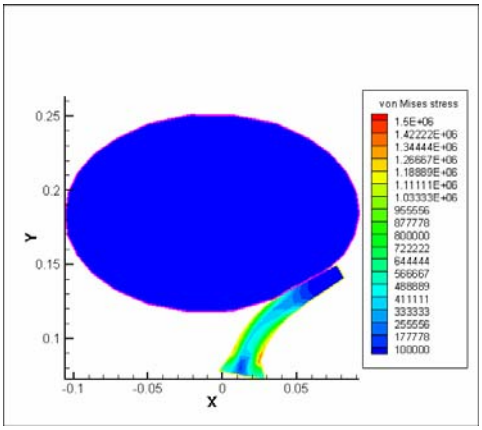
a) body stress distribution (aluminum)



b) finger stress distribution (aluminum)

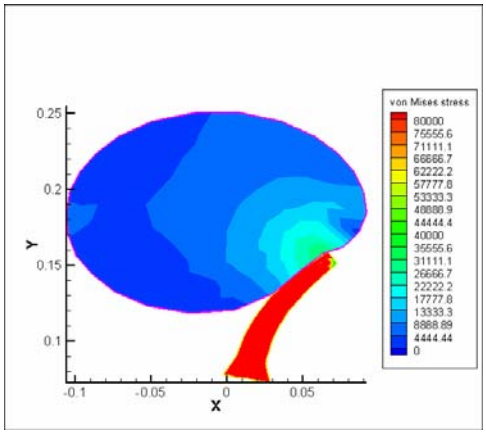


c) body stress distribution (rubber)

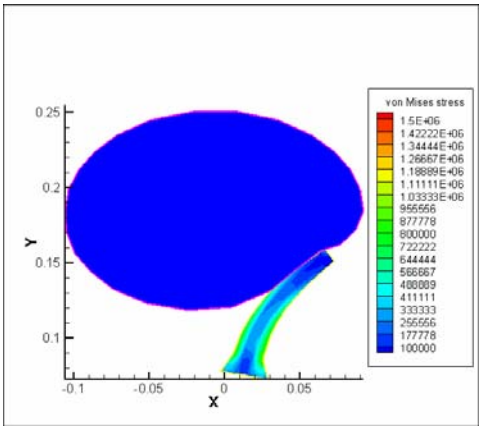


d) finger stress distribution (rubber)

Figure 6-25 contact stress distribution



e) body stress distribution (silicon)



f) finger stress distribution (silicon)

Figure 6-25 continued

It can be observed that the maximum stress in the body happens around the contact region. As the material becomes softer, the contact stress for both body and finger becomes smaller.

6.4 Summary

Two engineering application examples in live object handling research project are discussed in detail to illustrate the practical use of the meshless method. In the first example, the meshless method helped obtain the magnetic field distribution in the magnetic actuator such that the force characterization and the dynamic response of the actuator can be subsequently computed. In the second example, we construct a 2D flexible finger model for contact computation. The contact force obtained from the reduced 2D model closely matches with the result obtained from the FEM and those measured by experiment. Although two examples are from different physical disciplines, we show that the meshless method, as a general computational tool, is able to provide useful results to facilitate both system designs.

CHAPTER VII

CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

This research investigates practical issues related to the MLM and develops an adaptive algorithm to automatically insert additional nodes and improve computational accuracy. The study has been in the context of the engineering problems: magnetic field computation and large deformation contact. The contributions of this research can be briefly summarized as the followings:

A. Methods for handling the discontinuity in computational domain have been investigated for both strong-form and weak-form discretization methods.

When the computational domain is comprised of different materials, the discontinuity happens at the material boundary for the distribution of certain physical quantities, such as displacement derivative in mechanical problem, and magnetic flux intensity in magnetic field problem. Such discontinuity can be handled naturally in the linear FEM by aligning the edge of elements along the material boundary. However special treatment must be developed in MLM due to its higher order continuous basis function.

In strong form discretization, we solve this problem by dividing the computational domain into sub-domains based on their different material properties. At each nodal location at the material boundary, two nodes with different material properties are created to overlap with each other such that the discontinuity boundary condition can be

reinforced. One problem arising in applying this method to magnetic fields is that at the singular point of a material boundary such as corner, the normal of the boundary is undefined and thus, it is impossible to directly apply the boundary condition. We provide two solutions for this problem. In the first one, a pseudo normal is constructed by interpolating the normal obtained from both sides of the singular point. In the second one, we avoid the problem of normal by reformulating the boundary condition using the Gauss integral form of the Maxwell equation. Numerical investigation shows the second method achieves higher accuracy.

In the weak form discretization method, the ML basis function for the nodes at the material boundary is enriched with a discontinuous function such that the discontinuity is solved naturally and the computational domain can be computed as a whole. The comparison has been conducted for these proposed methods. The advantages and disadvantages of these methods have been illustrated by examples. The comparison shows that these methods can significantly improve the computational accuracy around the material field.

B. A relatively complete adaptive meshless computational scheme has been developed and implemented.

Without the limitation of FEM mesh, one of most attractive features of MLM is that it is relatively easier to insert additional nodes to the computational domain to improve the computational accuracy. In order to fully exploit this advantage of MLM, we develop a relatively complete adaptive computational algorithm which includes an error estimation method, a nodal insertion scheme and a method to improve numerical integration using the property of partition unity.

The adaptive computation correctly identifies the computational errors in the previous computational step. This provides a valuable means to insert additional nodes. Based on the idea of error estimation algorithms in FEM, we estimate the error in MLM using two simple steps: first, interpolating the previous result using ML basis function with different support size; then, estimating the error by computing the difference between the interpolated result and the original result.

For the nodal insertion scheme, a simple method based on the Voronoi diagram is used. The error is estimated at the vertex of each Voronoi cell. Additional nodes are inserted to locations with larger errors. To avoid the reconstruction of numerical integration cells after new nodes are inserted, numerical integration is performed by using a partition unity integration scheme rather than the popular background integration cells scheme.

The whole adaptive scheme is very easy to implement. However it is very effective in practice as shown by the given numerical examples.

The application to the magnetic field problem has shown that this method can be considered as a more general method to solve the discontinuity problem in magnetic fields. Without any special treatment at the material interface, large error regions around the material interface have been correctly identified, and additional nodes are inserted automatically. This has greatly simplified the computational procedure and reduced human involvement.

C. The adaptive meshless method is extended to solve the nonlinear large deformation contact problem.

In this extension, we handle the contact constraint by using the sliding line algorithm and the penalty method. We reformulate the error estimation algorithm to utilize the stress information rather than the displacement information. This modification has effectively eliminated the effect of rigid body motion and provided correct error estimation.

D. The developed method has been applied to practical engineering application.

We applied the MLM to two practical engineering applications encountered in the development of a live object handling system. In the first application, we developed a non-contact mechanical-magnetic actuator for manipulating the orientation of a bird. To satisfy the stringent timing requirement, simulations are conducted to obtain the dynamic response of the actuator for a given design configuration. For this, the developed MLM is applied to compute the magnetic fields of the actuator. The magnetic force is then calculated using the Lorenz force formulation. The final computed dynamic response of the actuator agrees with the experiment result very well.

In the second application, we apply the MLM to simulate the contact between a bird's body and a flexible finger. To avoid injuring the bird, a flexible finger that is capable of large deformation is used for grasping. In the previous study, a finger model based on flexible beam theory was proposed. This method, though able to predict the overall contact force exerted by flexible finger onto bird's body, can not provide the contact stress and strain information that ultimately determines damage to the bird. FEM has also been tried previously to attack this problem. However the excessively large deformation in this application has made FEM an inappropriate choice. In this study, we have provided a numerical example to demonstrate that the new method is able to

overcome the limitation of both the beam model and the FEM and is able to provide the detailed stress and strain information generated by the contact. This useful feature will facilitate the future design of the grasper.

7.2 Future Works

The current methods could be extended or further developed from the following perspectives.

1) In our mechanical-magnetic actuator, we did not use any material with nonlinear permeability such as iron. However such materials have been used in many other magnetic actuators to reduce the magnetic flux linkage and to increase magnetic force. Application of the meshless method to design such actuators requires further development of the current meshless method to handle nonlinearity of magnetic material.

2) Our current simulation for the mechanical-magnetic system ignored the effect of mechanical motion on magnetic field. Experiment result shows this simplification is acceptable for our situation. However this may result in unacceptable numerical error when the problem's scale is in the micro level. Considering such effects requires modification of the current dynamic formulation to enable a more seamless integration between electromagnetic computation and mechanical computation.

3) The contact algorithm developed in this thesis can be extended to simulate the material damage caused by contact. The phenomenon of cracking including its generation and expansion has been studied extensively in the context of fracture mechanics. However most previous research has focused on cracks induced by

external force applied in the far field. Furthermore, to avoid the difficulty of reconstructing geometry during the crack expansion, the previous computational methods often simplify the geometry of the crack into a line of discontinuity in the material, which is reasonable for engineering material with high toughness since the width of crack is often small in this case. Nevertheless, this simplification could generate larger numerical errors when softer materials, such as biomaterials, are involved. Methods to simulate the damage to soft materials induced by concentrated contact force have not been fully studied. The nature of the meshless method has made it easier to modify the geometry and create a new material interface when material damage happens. Our work provides a good starting point for developing a more general simulation method that is able to handle integratively the contact and the material damage.

4) The current method can be extended to simulate dynamic problem. For a dynamic problem, the simulation time may increase considerably, especially if the implicit formulation is used. The current implementation of the algorithms is in Matlab. It is relatively easier to test algorithms in Matlab, but it is not a good choice to simulate large scale computational job in Matlab due to its low-speed natural. A transfer from Matlab to a directly executable computer language such as C++ or FORTRAN is necessary. This transfer will also benefit the large scale static simulation.

REFERENCES

[1] K.-M. Lee, "On the development of a compliant grasping mechanism for on-line handling of live objects, Part I: Analytical model," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pp. 354-359, 1999.

[2] K.-M. Lee, "Design criteria for developing an automated live-bird transfer system," 2000, pp. 1138-1143, San Francisco, CA, USA.

[3] K.-M. Lee, R. Gogate, and R. Carey, "Automated singulating system for transfer of live broilers," 1998, pp. 3356-3361, Leuven, Belgium.

[4] K.-M. Lee and X. Yin, "Design algorithm for automated dynamic grasping of live birds," 2001, pp. 207-12, Como, Italy.

[5] K.-M. Lee, A. B. Webster, J. Joni, X. Yin, R. Carey, M. P. Lacy, and R. Gogate, "On the development of a compliant grasping mechanism for on-line handling of live objects, Part II: Design and experimental investigation," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pp. 360-365, 1999.

[6] X. Yin, K.-M. Lee, and C.-C. Lan, "Computational models for predicting the deflected shape of a non-uniform, flexible finger," 2004, pp. 2963-2968, New Orleans, LA, United States.

[7] K.-M. Lee, Z. Wei, and J. Joni, "Parametric study on pole geometry and thermal effects of a VRSM," 2005, pp. 548-53, Singapore.

[8] S. N. Atluri and T. Zhu, "A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics," *Computational Mechanics*, vol. 22, pp. 117-127, 1998.

[9] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, "Meshless methods: an overview and recent developments," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 3, 1996.

[10] C. A. Duarte and J. T. Oden, "h-p adaptive method using clouds," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 237, 1996.

[11] T. Belytschko, Y. Krongauz, M. Fleming, D. Organ, and W. K. Snm Liu, "Smoothing and accelerated computations in the element free Galerkin method," *Journal of Computational and Applied Mathematics*, vol. 74, pp. 111, 1996.

[12] T. Belytschko, Y. Y. Lu, and L. Gu, "Element-free Galerkin methods," *International Journal for Numerical Methods in Engineering*, vol. 37, pp. 229, 1994.

[13] I. Babuska and J. M. Melenk, "The Partition of Unity Method," *International Journal for Numerical Methods in Engineering*, vol. 40, pp. 727-758, 1997.

[14] N. Sukumar, B. Moran, and T. Belytschko, "The Natural Element Method in Solid Mechanics," *International Journal for Numerical Methods in Engineering*, vol. 43, pp. 839-887, 1998.

[15] W. Liu, S. Jun, and Y. F. Zhang, "Reproducing Kernel Particle Methods," *International Journal for Numerical Methods in Engineering*, vol. 20, pp. 1081-1106, 1995.

[16] Z. Wei, K.-M. Lee, T. S. W., Z. Zhou, and S.-P. Hong, "Free surface flow in high speed fiber drawing with large-diameter glass preforms," *Transactions of the ASME. Journal of Heat Transfer*, vol. 126, pp. 713-22, 2004.

[17] W. Liu, T. Belytschko, and H. Chang, "An arbitrary Lagrangian-Eulerian finite element method for path-dependent materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 58, pp. 227-246, 1986.

[18] W. Liu, H. Chang, J. Chen, and T. Belytschko, "Arbitrary Lagrangian and Eulerian Petrov-Gakerkin finite elements for nonlinear problem," *Computer Methods in Applied Mechanics and Engineering*, vol. 68, pp. 259-310, 1988.

[19] M. S. Gadala and J. Wang, "Practical procedure for mesh motion in arbitrary Lagrangian-Eulerian method," *Engineering with Computers*, vol. 14, pp. 223-234, 1998.

[20] P. O. Bouchard, F. Bay, Y. Chastel, and I. Tovena, "Crack propagation modelling using an advanced remeshing technique," *Computer Methods in Applied Mechanics and Engineering*, vol. 189, pp. 723-742, 2000.

[21] L. Lucy, "A numerical approach to testing the fission hypothsis," *Astron. J.*, vol. 8, pp. 1013-1024.

[22] R. A. a. M. Gingold, "Smooth particle hydrodynamics: theory and applications to nonspherical stars," *Mon. Not. R. Astron. Soc.*, pp. 375–389, 1977.

[23] J. J. Monaghan, "Smoothed Particale Hydrodynamics," *Annu. Rev. Astron. Astrophys.*, vol. 30, pp. 543-574, 1972.

[24] B. Nayroles, G. Touzot, and P. Villon, "Generalizing the finite element method: Diffuse approximation and diffuse elements," *Computational Mechanics*, vol. 10, pp. 307, 1992.

[25] W. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko, "Reproducing Kernel Particle Methods for Structural Dynamics," *International Journal for Numerical Methods in Engineering*, vol. 38, pp. 1655-1679, 1995.

[26] N. Sukumar, B. Moran, A. Semenov, and V. V. Belikov, "Natural Neighbor Galerkin Method," *International Journal for Numerical Methods in Engineering*, vol. 50, pp. 1-27, 2001.

[27] G. Yagawa and T. Yamada, "Free Mesh Method, A New Meshless Finite Element Method," *Computational Mechanics*, vol. 18, pp. 383-386, 1996.

[28] S. De and K. J. Bathe, "The Method of Finite Spheres," *Computational Mechanics*, vol. 25, pp. 329-345, 2000.

[29] B. N. Rao and S. Rahman, "Probabilistic fracture mechanics by Galerkin meshless methods - Part I: Rates of stress intensity factors," *Computational Mechanics*, vol. 28, pp. 351, 2002.

[30] Y. P. Chen, J. D. Lee, and A. Eskandarian, "Dynamic meshless method applied to nonlocal crack problems," *Theoretical and Applied Fracture Mechanics*, vol. 38, pp. 293, 2002.

[31] B. N. Rao and S. Rahman, "Mesh-free analysis of cracks in isotropic functionally graded materials," *Engineering Fracture Mechanics*, vol. 70, pp. 1, 2003.

[32] B. N. Rao and S. Rahman, "An enriched meshless method for non-linear fracture mechanics," *International Journal for Numerical Methods in Engineering*, vol. 59, pp. 197, 2004.

[33] M. Duflot and H. Nguyen-Dang, "A meshless method with enriched weight functions for fatigue crack growth," *International Journal for Numerical Methods in Engineering*, vol. 59, pp. 1945, 2004.

[34] Y. Chen, A. Eskandarian, M. Oskard, and J. D. Lee, "Meshless analysis of plasticity with application to crack growth problems," *Theoretical and Applied Fracture Mechanics*, vol. 41, pp. 83, 2004.

[35] X. Guo and D. Fang, "Analysis of piezoelectric fracture under combined mechanical and electrical loading based on meshless method," 2004, pp. 543, Sendai, Japan.

[36] M. Duflot and H. Nguyen-Dang, "Fatigue crack growth analysis by an enriched meshless method," *Journal of Computational and Applied Mathematics*, vol. 168, pp. 155, 2004.

[37] S. C. Fan, X. Liu, and C. K. Lee, "Enriched partition-of-unity finite element method for stress intensity factors at crack tips," *Computers and Structures*, vol. 82, pp. 445, 2004.

[38] S. C. Fan and N. Sheng, "Meshless formulation using NURBS basis functions for eigenfrequency changes of beam having multiple open-cracks," *Journal of Sound and Vibration*, vol. 269, pp. 781, 2004.

[39] I. V. Singh and R. Prakash, "The numerical solution of three dimensional transient heat conduction problems using element free Galerkin method," *Heat and Technology*, vol. 21, pp. 73, 2003.

[40] A. Eskandarian, Y. Chen, M. Oskard, and J. D. Lee, "Meshless Analyses of Fracture, Plasticity and Impact," 2003, pp. 89, Washington, DC, United States.

[41] B. N. Rao and S. Rahman, "Stochastic meshless analysis of elastic-plastic cracked structures," *Computational Mechanics*, vol. 32, pp. 199, 2003.

[42] B. N. Rao and S. Rahman, "A new interaction integral method for analysis of cracks in orthotropic functionally graded materials," 2003, pp. 215, Cleveland, OH, United States.

[43] L. L. Lou and P. Zeng, "FE-Meshless coupling method for 2D crack propagation," *Key Engineering Materials*, vol. 233-236, pp. 169, 2003.

[44] R. C. Batra and H. K. Ching, "Analysis of elastodynamic deformations near a crack/notch tip by the meshless local Petrov-Galerkin (MLPG) method," *Computer Modeling in Engineering & Sciences*, vol. 3, pp. 717, 2002.

[45] "New and emerging computational methods: Applications to fracture, damage, and reliability," 2002, pp. 246, Vancouver, BC, Canada.

[46] A. Carpinteri, G. Ferro, and G. Ventura, "Element-free crack propagation by partition of unity weighted quadrature," 2001, pp. 215, Vienna, Austria.

[47] A. Carpinteri, G. Ferro, and G. Ventura, "An augmented Lagrangian element-free (ALEF) approach for crack discontinuities," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 941, 2001.

[48] L. A. Barba, A. Leonard, and C. B. Allen, "Advances in viscous vortex methods - Meshless spatial adaption based on radial basis function interpolation," *International Journal for Numerical Methods in Fluids*, vol. 47, pp. 387, 2005.

[49] I. V. Singh, "The numerical solution of viscous fluid problems using meshless method," *Heat and Technology*, vol. 22, pp. 137, 2004.

[50] L. C. Felgueroso, I. Colominas, G. Mosqueira, F. Navarinna, and M. Casteleiro, "On the Galerkin formulation of the smoothed particle hydrodynamics method," *International Journal for Numerical Methods in Engineering*, vol. 60, pp. 1475, 2004.

[51] X. Zhou, Y. C. Hon, and K. F. Cheung, "A grid-free, nonlinear shallow-water model with moving boundary," *Engineering Analysis with Boundary Elements*, vol. 28, pp. 967, 2004.

[52] I. V. Singh, "Parallel implementation of the EFG method for heat transfer and fluid flow problems," *Computational Mechanics*, vol. 34, pp. 453, 2004.

[53] I. S. Raju, D. R. Phillips, and T. Krishnamurthy, "A radial basis function approach in the meshless local Petrov-Galerkin method for Euler-Bernoulli beam problems," *Computational Mechanics*, vol. 34, pp. 464, 2004.

[54] J. Guo and Z. Tao, "Modified moving particle semi-implicit meshless method for incompressible fluids," *Journal of Thermal Science*, vol. 13, pp. 226, 2004.

[55] I. V. Singh, "Application of meshless EFG method in fluid flow problems," *Sadhana*, vol. 29, pp. 285, 2004.

[56] S. R. Idelsohn, E. Onate, and F. Del Pin, "The particle finite element method: A powerful tool to solve incompressible flows with free-surfaces and breaking waves," *International Journal for Numerical Methods in Engineering*, vol. 61, pp. 964, 2004.

[57] I. Colominas, L. Cueto-Felgueroso, G. Mosqueira, F. Navarrina, and M. Casteleiro, "A particle numerical approach based on the SPH method for computational fluid mechanics," 2004, pp. 183, Bologna, Italy.

[58] M. A. Martinez, E. Cueto, I. Alfaro, M. Doblare, and F. Chinesta, "Updated Lagrangian free surface flow simulations with natural neighbour Galerkin methods," *International Journal for Numerical Methods in Engineering*, vol. 60, pp. 2105, 2004.

[59] A. Campo and B. Morrone, "Meshless approach for computing the heat liberation from annular fins of tapered cross section," *Applied Mathematics and Computation (New York)*, vol. 156, pp. 137, 2004.

[60] G. Maranzana, I. Perry, and D. Maillet, "Modeling of conjugate heat transfer between parallel plates separated by a hydrodynamically developed laminar flow by the quadrupole method," *Numerical Heat Transfer; Part A: Applications*, vol. 46, pp. 147, 2004.

[61] D. L. Young, S. C. Jane, C. Y. Lin, C. L. Chiu, and K. C. Chen, "Solutions of 2D and 3D stokes laws using multiquadrics method," *Engineering Analysis with Boundary Elements*, vol. 28, pp. 1233, 2004.

[62] H. Li, J. Q. Cheng, T. Y. Ng, J. Chen, and K. Y. Lam, "A meshless Hermite-Cloud method for nonlinear fluid-structure analysis of near-bed submarine pipelines under current," *Engineering Structures*, vol. 26, pp. 531, 2004.

[63] R. W. Johnson, "Development of a B-spline collocation method for CFD," 2003, pp. 47, New Orleans, LA, United States.

[64] G. Winter, J. C. Abderraman, J. A. Jimenez, B. Gonzalez, E. Benitez, and P. Cuesta, "Meshless numerical simulation of (full) potential flows in a nozzle by genetic algorithms," *International Journal for Numerical Methods in Fluids*, vol. 43, pp. 1167, 2003.

[65] D. Sridar and N. Balakrishnan, "An upwind finite difference scheme for meshless solvers," *Journal of Computational Physics*, vol. 189, pp. 1, 2003.

[66] M. A. Martinez, E. Cueto, M. Doblare, and F. Chinesta, "Natural element meshless simulation of flows involving short fiber suspensions," *Journal of Non-Newtonian Fluid Mechanics*, vol. 115, pp. 51, 2003.

[67] L. T. Zhang, G. J. Wagner, and W. K. Liu, "Modelling and simulation of fluid structure interaction by meshfree and FEM," *Communications in Numerical Methods in Engineering*, vol. 19, pp. 615, 2003.

[68] S. R. Idelsohn, E. Onate, and F. Del Pin, "A Lagrangian meshless finite element method applied to fluid-structure interaction problems," *Computers and Structures*, vol. 81, pp. 655, 2003.

[69] D. W. Pepper, C. S. Chen, and J. Li, "Modeling heat transfer using adaptive finite elements, boundary elements, and meshless methods," 2002, pp. 349, Kassandra, Greece.

[70] N. Uenoyama, G. Yagawa, and Y. Nakabayashi, "Computational fluid dynamics around pitching-bending wing using free mesh method," *Nippon Kikai Gakkai Ronbunshu, B Hen/Transactions of the Japan Society of Mechanical Engineers, Part B*, vol. 68, pp. 9, 2002.

[71] H. Lia, Q. X. Wang, and K. Y. Lani, "A variation of local point interpolation method (vLPIM) for analysis of microelectromechanical systems (MEMS) device," *Engineering Analysis with Boundary Elements*, vol. 28, pp. 1261, 2004.

[72] D. Xu, H. Ding, and C. Shu, "Application of local radial basis function-based differential quadrature method in micro flows," 2004, pp. 243, Boston, MA, United States.

[73] L. Hua, Q. X. Wang, and K. Y. Lam, "Development of a novel meshless Local Kriging (LoKriging) method for structural dynamic analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 2599, 2004.

[74] L. Gang and N. R. Aluru, "Efficient mixed-domain analysis of electrostatic MEMS," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, pp. 1228, 2003.

[75] G. Li and N. R. Aluru, "Linear, nonlinear and mixed-regime analysis of electrostatic MEMS," *Sensors and Actuators, A: Physical*, vol. 91, pp. 278, 2001.

[76] R. R. Ohs and N. R. Aluru, "Meshless analysis of piezoelectric devices," *Computational Mechanics*, vol. 27, pp. 23, 2001.

[77] V. Cingoski, N. Miyamoto, and H. Yamashita, "Element-free Galerkin method for electromagnetic field computations," *IEEE Transactions on Magnetics*, vol. 34, pp. 3236, 1998.

[78] K. Do Wan and K. Hong-Kyu, "Point collocation method based on the FMLSRK approximation for electromagnetic field analysis," *IEEE Transactions on Magnetics*, vol. 40, pp. 1029, 2004.

[79] S. L. Ho, S. Yang, J. M. Machado, and H. C. Wong, "Application of a meshless method in electromagnetics," *IEEE Transactions on Magnetics*, vol. 37, pp. 3198, 2001.

[80] X. Liang, Z. Zhiwei, B. Shanker, and L. Udpa, "Element-free Galerkin method for static and quasi-static electromagnetic field computation," *IEEE Transactions on Magnetics*, vol. 40, pp. 12, 2004.

[81] S. Liu, "Improvement of the element-free galerkin method for electromagnetic field calculation," *IEEE Transactions on Applied Superconductivity*, vol. 14, pp. 1866, 2004.

[82] Y. Marechal, "Some meshless methods for electromagnetic field computations," *IEEE Transactions on Magnetics*, vol. 34, pp. 3351, 1998.

[83] I. Nishiguchi and K. Haseyama, "Application of element-free Galerkin method to 2-D and 3-D magnetostatic problems," 2000, pp. 327, Pavia, Italy.

[84] Y. Shiyou, N. Guangzheng, J. R. Cardoso, S. L. Ho, and J. M. Machado, "A combined wavelet-element free Galerkin method for numerical calculations of electromagnetic fields," *IEEE Transactions on Magnetics*, vol. 39, pp. 1413, 2003.

[85] S. A. Viana and R. C. Mesquita, "Moving least square reproducing kernel method for electromagnetic field computation," *IEEE Transactions on Magnetics*, vol. 35, pp. 1372, 1999.

[86] W. K. Liu and Y. Chen, "Wavelet and multiple scale reproducing Kernel methods," *International Journal for Numerical Methods in Fluids*, vol. 21, pp. 901, 1995.

[87] G. R. Liu, "Mesh free methods: moving beyond the finite element method." Boca Raton, Fla.: CRC Press, 2003, pp. 692 p.

[88] E. Onate, S. Idelsohn, O. C. Zienkiewicz, and R. L. Taylor, "A finite point method in computational mechanics. Applications to convective transport and fluid flow," *International Journal for Numerical Methods in Engineering*, vol. 39, pp. 3839, 1996.

[89] H. A. Haus and J. R. Melcher, *Electromagnetic Fields and Energy*: Prentice Hall, 1989.

[90] A. A. Kostin, "The electrical image method in physical modelling of magnetic fields," *Elektrichestvo*, pp. 86, 1975.

[91] V. N. Beloozerov and M. L. Levin, "The image method in magnetostatics at a spherical superconducting boundary," *Zhurnal Tekhnicheskoi Fiziki*, vol. 36, pp. 3, 1966.

[92] O. C. Zienkiewicz, A. K. Bahrani, and P. L. Artett, "Solution of three-dimensional field problems by the finite element1 method," *The Engineer*, vol. 224, pp. 550, 1967.

[93] A. K. Chargin and C. D. Henning, "Magnetic field and force by finite element techniques," 1969, pp. 5 pp., Los Alamos, NM, USA.

[94] P. Silvester and M. V. K. Chari, "Finite element solution of saturable magnetic field problems," *IEEE Transactions on Power Apparatus and Systems*, vol. Pas-89, pp. 1642, 1970.

[95] J. H. Coggon, "Electromagnetic and electrical modeling by the finite element method," *Geophysics*, vol. 36, pp. 132, 1971.

[96] M. Ohta, N. Fujisawa, and S. Itoh, "Analysis of magnetic fields by the finite element method," *Transactions of the Institute of Electrical Engineers of Japan*, vol. 92, pp. 233, 1972.

[97] K. Dikshitulu, "Numerical techniques in the solution of field problems," *Bulletin of Electrical Engineering Education*, vol. 28, pp. 46, 1962.

[98] J. H. Collins and P. Daly, "Calculations for guided electromagnetic waves using finite-difference methods," *Journal of Electronics and Control*, vol. 14, pp. 380, 1963.

[99] H. Ching-Lai and F. Liang-Tseng, "A finite difference analysis of laminar magneto-hydrodynamic flow in the entrance region of a flat rectangular duct," *Applied Scientific Research, Section B (Electrophysics, Acoustics, Optics, Mathematical Methods)*, vol. 10, pp. 343, 1963.

[100] A. Wexler, "Some recent developments in field calculations," *IEEE Transactions on Magnetics*, vol. MAG-15, pp. 1659, 1979.

[101] M. H. Lean and A. Wexler, "Accurate field computation with the boundary element method," *IEEE Transactions on Magnetics*, vol. MAG-18, pp. 331, 1982.

[102] S. J. Salon and J. P. Peng, "A hybrid finite element-boundary element formulation of Poisson's equation for axisymmetric vector potential problems," *Journal of Applied Physics*, vol. 53, pp. 8420, 1982.

[103] M. A. Morjaria, S. Mukherjee, and F. C. Moon, "A boundary integral method for eddy current flow around cracks in thin plates," *IEEE Transactions on Magnetics*, vol. MAG-18, pp. 467, 1982.

[104] T. Nomura, S. Nakamura, and H. Kohmo, "Magnetic field calculation of superconducting alternators by a boundary element method," 1983, pp. 105, Hiroshima, Japan.

[105] N. Kikuchi and J. T. Oden, *Contact problems in elasticity: a study of variational inequalities and finite element methods*. Philadelphia: SIAM, 1988.

[106] E. Haug, R. Chand, and K. Pan, "Multibody elastic contact analysis by quadratic programming," *Journal of Optimization Theory and Applications*, vol. 21, pp. 189, 1977.

[107] A. R. Johnson and C. J. Quigley, "Frictionless geometrically nonlinear contact using quadratic programming," *International Journal for Numerical Methods in Engineering*, vol. 28, pp. 127, 1989.

[108] J. H. Heegaard and A. Curnier, "An augmented Lagrangian method for discrete large-slip contact problems," *International Journal for Numerical Methods in Engineering*, vol. 36, pp. 569, 1993.

[109] K. J. Bathe and A. Chaudhary, "A solution method for planar and axisymmetric contact problems," *International Journal for Numerical Methods in Engineering*, vol. 21, pp. 65, 1985.

[110] Z.-H. Zhong, *Finite element procedures for contact-impact problems*. Oxford; New York: Oxford University Press, 1993.

[111] J. C. Simo, P. Wriggers, and R. L. Taylor, "A perturbed Lagrangian formulation for the finite element solution of contact problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 50, pp. 163, 1985.

[112] J. C. Simo and T. A. Laursen, "An augmented Lagrangian treatment of contact problems involving friction," *Computers and Structures*, vol. 42, pp. 97, 1992.

[113] P. Wriggers, J. C. Simo, and R. L. Taylor, "Penalty and augmented Lagrangian fomulations for contact problems," 1985, pp. 97, Swansea, Wales.

[114] P. Wriggers, P. Scherf, and C. Carstensen, "An adaptive finite element model for frictionless contact problems," *Zeitschrift fur Angewandte Mathematik und Mechanik*, vol. 75, pp. 37, 1995.

[115] C. Carstensen, O. Scherf, and P. Wriggers, "Adaptive finite elements for elastic bodies in contact," *SIAM Journal of Scientific Computing*, vol. 20, pp. 1605, 1999.

[116] A. Rieger and P. Wriggers, "Adaptive methods for frictionless contact problems," *Computers and Structures*, vol. 79, pp. 2197, 2001.

[117] G. Hu and P. Wriggers, "On the adaptive finite element method of steady-state rolling contact for hyperelasticity in finite deformations," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 1333, 2002.

[118] P. Wriggers and O. Scherf, "Adaptive finite element techniques for frictional contact problems involving large elastic strains," *Computer Methods in Applied Mechanics and Engineering*, vol. 151, pp. 593-603, 1998.

[119] W. K. Liu and C. Oberste-Brandenburg, "Reproducing kernel and wavelet particle methods," 1993, pp. 39-55, New Orleans, LA, USA.

[120] W. K. Liu, J. Adee, S. Jun, and T. Belytschko, "Reproducing Kernel Particle Methods for elastic and plastic problems," 1993, pp. 175-189, New Orleans, LA, USA.

[121] J. S. Chen, C. Pan, C. M. O. L. Roque, and H. P. Wang, "Lagrangian reproducing kernel particle method for metal forming analysis," *Computational Mechanics*, vol. 22, pp. 289-307, 1998.

[122] P. Campbell and S. A. Al-Murshid, "The effect of the magnetization distribution within anisotropic alnico magnets upon field calculation," 1980, pp. 1032-4, Boston, MA, USA.

[123] L. W. Cordes and B. Moran, "Treatment of material discontinuity in the element-free Galerkin method," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 75, 1996.

[124] K.-M. Lee, Q. Li, and H. Sun, "Effects of numerical formulation on magnetic field computation using meshless methods," *IEEE Transactions on Magnetics*, vol. 42, pp. 2164-2171, 2006.

[125] O. C. Zienkiewicz and J. Z. Zhu, "Superconvergent patch recovery and a posteriori error estimates. Part 2: error estimates and adaptivity," *International Journal for Numerical Methods in Engineering*, vol. 33, pp. 1365, 1992.

[126] O. C. Zienkiewicz and J. Z. Zhu, "Superconvergent patch recovery and a posteriori error estimates. Part 1: the recovery technique," *International Journal for Numerical Methods in Engineering*, vol. 33, pp. 1331, 1992.

[127] A. Carpinteri, G. Ferro, and G. Ventura, "The partition of unity quadrature in meshless methods," *International Journal for Numerical Methods in Engineering*, vol. 54, pp. 987, 2002.

[128] N. I. J. P. A. Bastos, "Forces in Permanent Magnets Team Workshop Problem 23," *http://www.compumag.co.uk/team.html,* (10/2006).

[129] K.-M. Lee and H. Son, "Torque model for design and control of a spherical wheel motor," 2005, pp. 335, Monterey, CA, United States.

[130] L. Yan, I.-M. CHen, C. K. Lim, G. Yang, W. Lin, and K.-M. Lee, "Torque Modeling of a DC Spherical Actuator Based on Lorentz Force Law," International Conference on Control, Automation, Robotics and Vision 2002, Singapore.

[131] K. M. Lee and C. K. Kwan, "Design concept development of a spherical stepper for robotic applications," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 175, 1991.

[132] J. O. Hallquist, "A numerical treatment of sliding interfaces and impact," Computational Techniques for Interface Problems 1978, pp. 117-33, San Francisco, CA, USA.

[133] P. Wriggers, T. Vu Van, and E. Stein, "Finite element formulation of large deformation impact-contact problems with friction," *Computers and Structures*, vol. 37, pp. 319-31, 1990.

[134] Q. Li and K. M. Lee, "An adaptive meshless method for magnetic field computation," *IEEE Transactions on Magnetics*, vol. In press, 2006.

[135] L. E. Malvern, *Introduction to the mechanics of a continuous medium [by] lawrence e. malvern*. Englewood Cliffs, N.J., Prentice-hall [1969], 1969.

[136] W. K. Liu, S. Jun, and Y. F. Zhang, "Reproducing kernel particle methods," *International Journal for Numerical Methods in Fluids*, vol. 20, pp. 1081, 1995.

[137] K. L. Johnson, *Contact mechanics*. Cambridge [Cambridgeshire]; New York: Cambridge University Press, 1985.

[138] K.-M. Lee, "Design criteria for developing an automated live-bird transfer system," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 483-90, 2001.

[139] Q. Li and K.-M. Lee, "Effects of color characterization on computational efficiency of feature detection with live-object handling applications," 2005, pp. 225-30, Monterey, CA, USA.

[140] X. Yin, "Design and analysis of a compliant grasper for handling live objects," Georgia Institute of Technology, Atlanta, Ph. D. Thesis 2004.

[141] J. Joni, "Quasi-static force analysis of an automated live-bird transfer system," Georgia Institute of Technology, Atlanta, Master Thesis 2001.